

LREC 2026

The Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026)

Workshop Proceedings

Editors
Kengatharaiyer Sarveswaran and Ashwini Vaidya

16 May 2026

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026)

©ELRA Language Resources Association (ELRA), 2026
These proceedings are licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0)

ISBN 978-2-493814-66-1

Message from the Organising Chairs

Welcome to the proceedings of CHiPSAL 2026, the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL), held as part of the Language Resources and Evaluation Conference (LREC 2026). Building on the success of the inaugural workshop at COLING 2025, CHiPSAL 2026 continues to provide a platform for addressing the unique challenges of processing South Asian languages and for fostering collaboration among researchers working in this space.

South Asia is one of the most linguistically diverse regions in the world, yet its languages remain significantly underrepresented in language technologies, particularly in the era of large language models. This workshop brings together contributions that address key challenges such as data scarcity, morphological richness, script diversity, code-mixing, and the need for culturally grounded evaluation. The accepted papers span a wide range of languages, including both widely spoken languages and extremely low-resource and endangered languages such as Burushaski, Limbu, and Nepal Bhasha (Newari), reflecting the breadth and depth of current research efforts in the region.

We received 42 submissions to the main track, of which 21 papers were accepted for presentation, resulting in an acceptance rate of approximately 50%. Following withdrawals, 19 papers are included in the final proceedings. All submissions underwent a rigorous peer-review process, with each paper reviewed by three programme committee members. In total, 55 programme committee members from academia and industry worldwide contributed to the review process. In addition, the shared task received 15 system description paper submissions, of which 12 were accepted for publication.

CHiPSAL 2026 featured regular papers and a shared task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes for Nepali, which attracted strong participation from the research community. The diversity of submissions and the range of approaches highlight both the progress made and the challenges that remain in developing inclusive and effective NLP systems for South Asian languages.

We would like to express our sincere gratitude to the programme committee members for their timely and rigorous reviews. We also thank all authors for their valuable contributions and the LREC 2026 workshop chairs and organising committee for their support in facilitating this workshop.

We congratulate all authors of accepted papers and shared task contributions. We hope that CHiPSAL 2026 serves as a valuable forum for exchanging ideas, building collaborations, and advancing research on South Asian language technologies. We also look forward to expanding this initiative through future editions and regional events to further strengthen the research community in this domain.

For more information and future communications, please visit the workshop website: https://sites.google.com/view/chipsal/.

Organising Chairs of CHiPSAL 2026
Kengatharaiyer Sarveswaran, University of Jaffna, Jaffna, Sri Lanka.
Ashwini Vaidya, Indian Institute of Technology Delhi, India.
Bal Krishna Bal, Kathmandu University, Kathmandu, Nepal.
Surendrabikram Thapa, Virginia Tech, USA.
Tafseer Ahmed Khan, Mohammad Ali Jinnah University, Karachi, Pakistan.

Organising Committee

Organising Chairs
Kengatharaiyer Sarveswaran, University of Jaffna, Sri Lanka
Ashwini Vaidya, Indian Institute of Technology Delhi, India
Bal Krishna Bal, Kathmandu University, Nepal
Surendrabikram Thapa, Virginia Tech, USA
Tafseer Ahmed Khan, Mohammad Ali Jinnah University, Pakistan

Program Committee Members (alphabetical order)
A. M. Abirami, Thiagarajar College of Engineering (TCE), India
Ahrane Mahaganapathy, University of Jaffna, Sri Lanka
Aloka Fernando, IIT Colombo, Sri Lanka
Amba Kulkarni, University of Hyderabad, India
Aman Shakya, Pulchowk Campus, Tribhuvan University, Nepal
Ann Sinthusha Anton Vijeevaraj, University of Vavuniya, Sri Lanka
Arjuna S. R., Manipal Academy of Higher Education (MAHE), Bengaluru, India
Arafat Ahsan, IIIT Hyderabad, India
Ashwini Vaidya, IIT Delhi, India
Bal Krishna Bal, Kathmandu University, Nepal
Balaram Prasain, Central Department of Linguistics, Tribhuvan University, Nepal
Basanta Joshi, Pulchowk Campus, Tribhuvan University, Nepal
Buddhika Karunarathne, University of Moratuwa, Sri Lanka
Eugene Y. A. Charles, University of Jaffna, Sri Lanka
Faisal Alvi, Habib University, Pakistan
Farah Adeeba, University of Engineering and Technology, Pakistan
Gihan Dias, University of Moratuwa, Sri Lanka
Hariram Veeramani, UCLA, USA
Hassan Sajjad, Dalhousie University, Canada
Kritesh Rauniyar, Macquarie University, Australia
Krishna Chalise, Central Department of Linguistics, Tribhuvan University, Nepal
Lakshmojee Koduru, KPMG LLP, USA
Lekhnath Sharma Pathak, Central Department of Linguistics, Tribhuvan University, Nepal
Mazhar Ali, Benazir Bhutto Shaheed University Lyari, Pakistan
Meesum Alam, Indiana University, USA
Menan Velayuthan, Utrecht University, Netherlands
M. S. V. P. J. Sathvik, University of Birmingham, United Kingdom
Muhammad Rafi, FAST-NUCES, Pakistan
Muralikrishna S. N., MIT-MAHE, India
Munief Tahir, Al-Khawarizmi Institute of Computer Science, Pakistan
Mutee Ur Rahman, ISRA University, Pakistan
Naveed Unjum, FAU Erlangen-Nürnberg, Germany
Nisansa de Silva, University of Moratuwa, Sri Lanka
Parameswari Krishnamurthy, IIIT Hyderabad, India
Poonam Bansal, IGDTUW, India
Prakash Poudyal, Kathmandu University, Nepal
Rajani Chulyadyo, Kathmandu University, Nepal
Randil Pushpananda, University of Colombo School of Computing, Sri Lanka
Ritesh Kumar, UnReaL-TecE LLP
Sahar Rauf, Al-Khawarizmi Institute of Computer Science, Pakistan
Sajjad Haider, Institute of Business Administration, Pakistan

Sana Shams, Al-Khawarizmi Institute of Computer Science, Pakistan
Sanjukta Ghosh, IIT BHU, India
Shuvam Shiwakoti, Virginia Tech, USA
Siddhant Bikram Shah, Northeastern University, USA
Suganya Ramamoorthy, Vellore Institute of Technology, India
Supriya Khadka, Coventry University, United Kingdom
Surabhi Adhikari, Columbia University, USA
Surangika Ranathunga, Massey University, New Zealand
Surendrabikram Thapa, Virginia Tech, USA
Tafseer Ahmed, Mohammad Ali Jinnah University, Pakistan
Toqeer Ehsan, VTT Technical Research Centre of Finland
Usman Naseem, Macquarie University, Australia
Uthayasanker Thayasivam, University of Moratuwa, Sri Lanka
Wazir Ali, Quaid-e-Awam University of Engineering, Science and Technology, Pakistan

Volunteers (alphabetical order)
Ahrane Mahaganapathy, University of Jaffna, Sri Lanka
Ajintha Sivakulasingam, University of Jaffna, Sri Lanka
Nishanthini Kanthakumar, University of Jaffna, Sri Lanka

Table of Contents

Findings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026)
Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed and Bal Krishna Bal 1

Development of Burushaski Speech - English Text Translation Dataset
Tauqeer Saleem, Abdul Samad, Azkaa Nasir, Adina Adnan Mansoor, Fatima Faisal and Mahrukh Yousuf 9

Lost the Negation or Lost in Negation
Vennela Bairi and Parameswari Krishnamurthy 18

A Morphological Transducer for the Limbu Language
Avyaya Singh and Jonathan N. Washington 32

Evaluating Large Language Models for Medical Named Entity Recognition in Urdu: A Benchmark Study
Bushra Nasim, Kinza Latif, Muhammad Zohair, Muhammad Hassan Asif and Zarmeen Nasim 40

DR-RAG: Addressing Retrieval Misalignment in Low-Resource Urdu Question Answering
Saad Ahmad, Muhammad Hammad, Muhammad Zeeshan, Faizad Ullah and Asim Karim 49

Cross-Domain Evaluation of Transformer-Based Models for Punjabi Speech Emotion Recognition
Fatima Tu Zahra, Kulsoom Asim, Sandesh Kumar and Abdul Samad 59

SiPaKosa: A Comprehensive Corpus of Canonical and Classical Buddhist Texts in Sinhala and Pali
Ranidu Hansaka Gurusinghe and Nevidu Jayatilleke 68

BNLI: A Linguistically-Refined Bengali Dataset for Natural Language Inference
Farah Binta Haque, Md Yasin, Shishir Saha, Md Shoaib Akhter Rafi and Farig Sadeque85

Exploring Large Language Models for Multitask Learning in Bengali Text Classification
Md. Sajjad Hossain, Kawsar Ahmed, SUNY MD ASHRAF KHAN and Mohammed Moshiul Hoque 92

Nwāchā Munā: A Devanagari Speech Corpus and Proximal Transfer Benchmark for Nepal Bhasha ASR
Rishikesh Kumar Sharma, Safal Narshing Shrestha, Jenny Poudel, Rupak Tiwari, Arju Shrestha, Rupak Raj Ghimire and Bal Krishna Bal 105

From Romanized to Devanagari: Enhancing Nepali Sentiment Analysis with NepaliXlit
Suraj Patel, Kashish Kumari Dhami, Norden Sherpa and Supriya Khadka 115

Why Does Low-Rank Adaptation Work for Hindi-English Code-Mixing? A Geometric Analysis
Shashank Vishwakarma and Rakesh Kumar 127

Hi-SEMFLOW: Lie Algebra–Based Semantic Flow for Span-Level Informal Language Identification in Hindi
Manikandan Ravikiran, Tanmay Tiwari, Vibhu Gupta and Rohit Saluja 137

NeCCo: Nepali Cultural Commonsense Benchmark for Large Language Model Evaluation
Sanket Shrestha, Raunak Regmi, Sadikshya Ghimire, Satyam Rana and Supriya Khadka 154

Reward-Guided Fine-Tuning of Whisper for Low-Resource Nepali Speech Recognition
Aadarsh Pandit, Yudhin Khanal, Ishan Pandey, Kushal Kunwar and Sunil Regmi 169

Evaluating Linguistic Knowledge of LLMs in Tamil: The ILAKKANAM Benchmark
Jeyarajalingam Varsha, Menan Velayuthan, Sumirtha Karunakaran, Rasan Nivethiga and Kengatharaiyer Sarveswaran 176

A Feature-Fusion Ensemble Approach for Tamil Hate Speech Detection
Sathasivam Nerujan and Kengatharaiyer Sarveswaran 190

Comparative Analysis of Tokenizers in Tamil Text Classification in Low Resource Settings
Gokulan Sivakumaran, Randil Pushpananda and 198

Improving Public Health Safety in Low-Resource Languages Using a Human-Verified Health Misinformation Corpus and Large Language Models
Sujal Maharjan, Astha Shrestha, Laxmi Thapa, Sweta Poudel, Shuvam Shiwakoti, Rabin Thapa, Kritesh Rauniyar and Surendrabikram Thapa 209

Multimodal Hate and Sentiment Understanding in Low-Resource Text-Embedded Images for Online Safety and Digital Well-being
Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal and Usman Naseem 219

Unigoa@CHiPSAL 2026: Early vs Late Fusion for Multimodal Hate and Sentiment Detection in Nepali Memes
Ashweta Fondekar, Milind Shivolkar and Jyoti Pawar 229

HasNat@CHiPSAL 2026: Multimodal Hate Speech Detection in Low-Resource Nepali Memes Using Aligned Vision–Language Models
Alvee Hasan Chowdhury, MD. ABUL HASNAT and Adnan Faisal 237

TeamHerald@CHIPSAL 2026: Hate Speech Detection and Sentiment Analysis of Nepali Memes Using Transformer-based Architectures and Ensemble Learning
Ashish Acharya, Anish Khatiwada, Rohit Khadka and Pragya Aryal 244

NeuralNoodles@CHiPSAL 2026: Late-Fusion Multimodal Stacking for Nepali Meme Sentiment Classification
Sidratul Muntaha, Sabila Anzum, Arpita Mallik and Hasan Murad 250

Multi-Modal-Minds@CHiPSAL 2026: A Comparative Study of Textual, Visual and Multimodal Architecture for Nepali Meme Moderation
Sandesh Shrestha, Bikram K.C., Akshyat Shah, Ashish Acharya and Rabin Thapa . . . 258

linus@CHiPSAL 2026: Multimodal Hate Speech and Sentiment Detection in Low-Resource Memes Using Late-Fusion Hybrid Architecture
Sunil Regmi, Bipesh Subedi, Saugat Singh and Suman Shrestha 267

ZeroR@CHiPSAL 2026: Two-Stage Vision-Language Adaptation with Contrastive Learning for Nepali Meme Classification
Nitiz Khanal 275

Team Oryu@CHiPSAL 2026: Integrating Text and Vision Transformers for Multimodal Hate Speech Detection in Memes
Noore Tamanna Orny, Joyeta Barua Moni, Md. Abtahee Kabir and Hasan Murad 284

Cuet Yet Another Baseline@CHiPSAL LREC 2026: Shared Task on Multimodal Sentiment Understanding in Low-Resource Memes
Rotna Dipika Debnath, Shahrin Afroz Hoque Ruhi, Ayesha Labiba, Arpita Mallik and Hasan Murad 292

MEME-Fusion@CHiPSAL 2026: Multimodal Ablation Study of Hate Detection and Sentiment Analysis on Nepali Memes
Samir Wagle, Reewaj Khanal and Abiral Adhikari 300

eGrantha.ai@CHiPSAL 2026: Stochastic Image Captioning for Robust Hate Speech Detection in Low-Resource Nepali Memes
Anish Thapaliya 308

EthosAI@CHiPSAL2026: Hate and Sentiment Understanding in Low-Resource Memes Using a Multimodal Approach
Vinayak Bansal, Deepawali Sharma, Aakash Singh and Vivek Kumar Singh 316

CHiPSAL2026 Workshop Program

16 May 2026

14:00 **Workshop overview**

Findings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026)
Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed and Bal Krishna Bal

14:15–14:40 **Keynote address by Dr Usman Naseem, Macquarie University, Australia**

14:40 **Session 1**
Chair: Professor Miriam Butt

Development of Burushaski Speech - English Text Translation Dataset
Tauqeer Saleem, Abdul Samad, Azkaa Nasir, Adina Adnan Mansoor, Fatima Faisal and Mahrukh Yousuf

Lost the Negation or Lost in Negation
Vennela Bairi and Parameswari Krishnamurthy

A Morphological Transducer for the Limbu Language
Avyaya Singh and Jonathan N. Washington

Evaluating Large Language Models for Medical Named Entity Recognition in Urdu: A Benchmark Study
Bushra Nasim, Kinza Latif, Muhammad Zohair, Muhammad Hassan Asif and Zarmeen Nasim

DR-RAG: Addressing Retrieval Misalignment in Low-Resource Urdu Question Answering
Saad Ahmad, Muhammad Hammad, Muhammad Zeeshan, Faizad Ullah and Asim Karim

Cross-Domain Evaluation of Transformer-Based Models for Punjabi Speech Emotion Recognition
Fatima Tu Zahra, Kulsoom Asim, Sandesh Kumar and Abdul Samad

SiPaKosa: A Comprehensive Corpus of Canonical and Classical Buddhist Texts in Sinhala and Pali
Ranidu Hansaka Gurusinghe and Nevidu Jayatilleke

BNLI: A Linguistically-Refined Bengali Dataset for Natural Language Inference
Farah Binta Haque, Md Yasin, Shishir Saha, Md Shoaib Akhter Rafi and Farig Sadeque

16 May 2026 (continued)

Exploring Large Language Models for Multitask Learning in Bengali Text Classification
Md. Sajjad Hossain, Kawsar Ahmed, SUNY MD ASHRAF KHAN and Mohammed Moshiul Hoque

16:00–16:30 **Keynote address by Professor Monojit Choudhury - "The Curious Case of Honorifics in South Asian Languages and Their Treatment in LLMs"**

16:30 **Session 2**
Chair: Professor Monojit Choudhury, Mohamed bin Zayed University of Artificial Intelligence

Nwāchā Munā: A Devanagari Speech Corpus and Proximal Transfer Benchmark for Nepal Bhasha ASR
Rishikesh Kumar Sharma, Safal Narshing Shrestha, Jenny Poudel, Rupak Tiwari, Arju Shrestha, Rupak Raj Ghimire and Bal Krishna Bal

From Romanized to Devanagari: Enhancing Nepali Sentiment Analysis with NepaliXlit
Suraj Patel, Kashish Kumari Dhami, Norden Sherpa and Supriya Khadka

Why Does Low-Rank Adaptation Work for Hindi-English Code-Mixing? A Geometric Analysis
Shashank Vishwakarma and Rakesh Kumar

Hi-SEMFLOW: Lie Algebra–Based Semantic Flow for Span-Level Informal Language Identification in Hindi
Manikandan Ravikiran, Tanmay Tiwari, Vibhu Gupta and Rohit Saluja

NeCCo: Nepali Cultural Commonsense Benchmark for Large Language Model Evaluation
Sanket Shrestha, Raunak Regmi, Sadikshya Ghimire, Satyam Rana and Supriya Khadka

Reward-Guided Fine-Tuning of Whisper for Low-Resource Nepali Speech Recognition
Aadarsh Pandit, Yudhin Khanal, Ishan Pandey, Kushal Kunwar and Sunil Regmi

16 May 2026 (continued)

17:15 **Session 3**
Chair: TBC

Evaluating Linguistic Knowledge of LLMs in Tamil: The ILAKKANAM Benchmark
Jeyarajalingam Varsha, Menan Velayuthan, Sumirtha Karunakaran, Rasan Nivethiga and Kengatharaiyer Sarveswaran

A Feature-Fusion Ensemble Approach for Tamil Hate Speech Detection
Sathasivam Nerujan and Kengatharaiyer Sarveswaran

Comparative Analysis of Tokenizers in Tamil Text Classification in Low Resource Settings
Gokulan Sivakumaran, Randil Pushpananda and Bandara

Improving Public Health Safety in Low-Resource Languages Using a Human-Verified Health Misinformation Corpus and Large Language Models
Sujal Maharjan, Astha Shrestha, Laxmi Thapa, Sweta Poudel, Shuvam Shiwakoti, Rabin Thapa, Kritesh Rauniyar and Surendrabikram Thapa

17:45 **Findings - Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes@CHiPSAL2026**

Multimodal Hate and Sentiment Understanding in Low-Resource Text-Embedded Images for Online Safety and Digital Well-being
Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal and Usman Naseem

17:55 **Concluding remarks**

Findings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026)

Kengatharaiyer Sarveswaran[1], Surendrabikram Thapa[2], Ashwini Vaidya[3], Tafseer Ahmed[4], Bal Krishna Bal[5]

[1]University of Jaffna, Sri Lanka, [2]Virginia Tech, USA, [3]Indian Institute of Technology Delhi, India, [4]Mohammad Ali Jinnah University, Pakistan, [5]Kathmandu University, Nepal

Abstract

This paper presents the findings of the second workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026), held as part of LREC 2026. South Asia is one of the most linguistically diverse regions in the world, yet its languages remain severely underrepresented in language resources and technologies, particularly in the era of large language models (LLMs). The workshop brings together research addressing key challenges in this space, including data scarcity, morphological complexity, code-mixing, script diversity, and the lack of culturally grounded evaluation benchmarks. The workshop received 57 submissions, covering a wide range of languages, tasks, and modalities, including both widely spoken languages (e.g., Bengali, Hindi, Tamil, and Urdu) and extremely low-resource and endangered languages such as Burushaski, Limbu, and Nepal Bhasha (Newari). Several contributions introduce arguably first-of-their-kind resources and benchmarks for these languages, spanning both text and speech domains, and focusing on linguistically informed and culturally grounded data creation. In addition to the main track, the workshop hosted a shared task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes for Nepali, attracting strong community participation. The results highlight the effectiveness of multimodal approaches while also revealing persistent challenges in modelling culturally nuanced and low-resource data. Across the accepted papers and shared task, key insights include the central role of high-quality data, the limitations of current multilingual models in low-resource settings, and the need for culturally aware and data-centric approaches. Overall, CHiPSAL 2026 demonstrates the growing momentum in South Asian language processing and highlights the importance of sustained, community-driven efforts to build inclusive and representative language technologies.

Keywords: South Asian languages, low-resource NLP, linguistic resources, multimodal learning, shared task

1. Introduction

South Asia—comprising Afghanistan, Bangladesh, Bhutan, India, Maldives, Nepal, Pakistan, and Sri Lanka—is home to nearly one-quarter of the world's population and represents one of the most linguistically diverse regions globally. The region includes languages from at least five major language families, along with several putative isolates, and comprises around 700 languages (Hock and Bashir, 2016). These languages are written in more than 25 scripts, reflecting a rich and complex linguistic ecology. In addition, a global diaspora exceeding 50 million speakers further extends the use of South Asian languages beyond the region. Despite this diversity and demographic significance, South Asian languages remain severely underrepresented in language resources and technologies.

Recent advances in large language models (LLMs) have led to significant progress in natural language processing; however, South Asian languages remain marginally represented in these models due to the limited availability of high-quality, curated data. While Unicode standardisation has largely addressed encoding issues, challenges persist in rendering, normalisation, and input methods, particularly for complex scripts and low-resource settings. Furthermore, the linguistic characteristics of South Asian languages—including rich morphology (Sarveswaran et al., 2021), diverse writing systems, code-mixing, and strong dialectal variation—pose substantial challenges for NLP. Closely related languages and shared scripts further complicate modelling and evaluation, often leading to performance disparities and unintended biases in multilingual systems.

The CHiPSAL (Challenges in Processing South Asian Languages) workshop addresses these challenges by bringing together research that focuses on linguistic, cultural, and technological aspects of processing South Asian languages. In particular, it emphasises the development of high-quality linguistic resources, culturally grounded evaluation benchmarks, and methods tailored to low-resource settings. By fostering collaboration across diverse language communities and research groups, the workshop aims to advance the state of NLP for South Asian languages while supporting the preservation and computational inclusion of their linguistic and cultural heritage. In this paper, we provide an overview of the accepted papers, shared tasks, and key directions emerging from the workshop.

This workshop represents the second edition of the CHiPSAL series, building on the success of the first workshop (Sarveswaran et al., 2025), which was co-located with COLING 2025 and brought to-

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 1–8
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

gether a growing community of researchers and practitioners working on South Asian language technologies.

2. Submission and Review processes

The workshop proposal was accepted by the LREC workshop chairs, after which the call for papers was disseminated through various channels, including mailing lists, direct contact emails, and the official workshop website[1]. In addition to the main workshop, a shared task was organised to further engage the research community and encourage broader participation and contributions towards South Asian language processing.

We received a total of 42 submissions to the main track of the CHiPSAL workshop. Of these, 2 papers were withdrawn by the authors, and the remaining submissions were considered for review. Following the review process, 21 papers were accepted for presentation, resulting in an acceptance rate of approximately 50%. Subsequently, 2 accepted papers were withdrawn, and 19 papers are included in the final proceedings.

All submissions underwent a rigorous peer-review process, with each paper evaluated by three program committee members to ensure a fair and thorough assessment. In total, 55 program committee members from academia and industry worldwide contributed to the review process. Among the accepted papers, selections for presentation were made to ensure coverage of diverse tasks and languages while accommodating the workshop schedule. The accepted papers were presented as oral presentations within the allocated time.

In addition to the main track, the workshop hosted a shared task on "Multimodal Hate and Sentiment Understanding in Low-Resource Memes," comprising two subtasks: (1) hate speech detection and (2) sentiment analysis. The shared task attracted participation from 23 teams, resulting in 14 system description paper submissions, of which 12 papers were accepted for publication.

3. Overview of Submissions

The workshop includes several contributions that extend computational research to extremely underrepresented and endangered languages by introducing foundational resources and tools. These include a structured Burushaski–English speech translation dataset comprising 10 hours of curated audio from 42 speakers, designed for an oral language with complex morphology (Saleem et al., 2026); the first morphological transducer for Limbu, enabling computational analysis for an endangered Sino-Tibetan language (Singh and Washington, 2026); and a 5.39-hour annotated speech corpus with an accompanying ASR benchmark for Nepal Bhasha (Newari), establishing initial infrastructure for speech technology in this language (Sharma et al., 2026). These works collectively contribute new linguistic coverage and resource development in areas that have remained largely absent from prior NLP research.

The papers present a substantial body of work on the creation of high-quality, task-specific datasets and benchmarks that are explicitly grounded in linguistic structure and cultural context, moving beyond opportunistic or purely scraped corpora. BNLI introduces a linguistically curated Bengali NLI dataset designed to address annotation errors, ambiguity, and lack of diversity through a controlled annotation pipeline (Haque et al., 2026)[2] . Similarly, NEGTEG provides a structured benchmark for Telugu comprising five tasks—negation detection, translation, paraphrase detection, sentiment analysis, and polarity flipping—enabling systematic evaluation of a core linguistic phenomenon (Bairi and Krishnamurthy, 2026)[3] . NeCCo extends this direction by introducing a culturally grounded Nepali commonsense benchmark spanning five domains, including kinship, rituals, and idiomatic expressions, thereby incorporating culturally embedded knowledge into evaluation (Shrestha et al., 2026b)[4] . In the Tamil context, ILAKKANAM constructs a benchmark of 820 linguistically annotated school-level questions across multiple linguistic categories, enabling fine-grained evaluation of linguistic competence (Varsha et al., 2026). Complementing these, Nep-Health-Misinfo provides a human-verified corpus for health misinformation detection in Nepali, constructed through a machine translation post-editing pipeline involving native experts (Maharjan et al., 2026)[5].

In addition to task-specific benchmarks, the workshop also introduces large-scale and diverse corpora that significantly expand data availability for low-resource languages across modalities and domains. SiPaKosa contributes a substantial Sinhala–Pali corpus comprising 786K sentences and 9.25M words drawn from 16 historical Buddhist documents and canonical sources, combining OCR-processed manuscripts with curated web data (Gursinghe and Jayatilleke, 2026)[6]. For Nepali, a Romanized social media dataset of over 6,500 comments, including a balanced subset of 1,518 manually annotated instances, supports the study

[1] https://sites.google.com/view/chipsal/

[2] BNLI-Dataset

[3] NEGTEG Dataset

[4] NeCCo Dataset

[5] Nep-Health-Misinfo Dataset

[6] SiPaKosa Dataset

Language	Language Family	Papers
Bengali	Indo-Aryan	(Haque et al., 2026; Hossain et al., 2026)
Burushaski	Language Isolate	(Saleem et al., 2026)
Hindi	Indo-Aryan	(Ravikiran et al., 2026; Vishwakarma and Kumar, 2026)
Limbu	Sino-Tibetan	(Singh and Washington, 2026)
Nepal Bhasha (Newari)	Sino-Tibetan	(Sharma et al., 2026)
Nepali	Indo-Aryan	(Shrestha et al., 2026b; Patel et al., 2026; Maharjan et al., 2026; Pandit et al., 2026)
Pali	Indo-Aryan	(Gururangsinghe and Jayatilleke, 2026)
Punjabi	Indo-Aryan	(Zahra et al., 2026)
Sinhala	Indo-Aryan	(Gururangsinghe and Jayatilleke, 2026)
Tamil	Dravidian	(Nerujan and Sarveswaran, 2026; Sivakumaran et al., 2026; Varsha et al., 2026)
Telugu	Dravidian	(Bairi and Krishnamurthy, 2026)
Urdu	Indo-Aryan	(Nasim et al., 2026; Ahmad et al., 2026)

Table 1: Languages covered in CHiPSAL 2026 regular (non–shared task) papers.

of cross-script and informal language processing (Patel et al., 2026)[7]. Speech resources are also represented: a Punjabi speech emotion dataset covering four emotion classes is introduced alongside a multi-strategy evaluation framework (Zahra et al., 2026)[8], while additional datasets include a publicly released Nepali ASR corpus used for reward-guided training (Pandit et al., 2026)[9] and an Urdu news headline dataset annotated for medical named entity recognition (Nasim et al., 2026)[10] .

4. Summaries of Accepted Papers

The BNLI dataset addresses limitations in Bengali NLI resources such as annotation errors, ambiguity, and lack of diversity. It is constructed with a rigorous annotation pipeline ensuring semantic clarity and balanced entailment, contradiction, and neutrality classes. Benchmarking with multilingual and Bengali-specific transformer models demonstrates improved reliability and interpretability, establishing BNLI as a strong foundation for Bengali and other low-resource inference tasks (Haque et al., 2026).

For Tamil hate speech detection, a hybrid framework combines L3Cube-TamilBERT representations with FastText embeddings to address subword fragmentation. Using Last-4 layer averaging and dual pooling (mean + max), the model achieves a Macro-F1 of 0.7883 and Hate Recall of 0.7503. A stacking ensemble further achieves Hate Precision of 0.9296, demonstrating improved detection performance (Nerujan and Sarveswaran, 2026).

The Burushaski–English speech translation dataset introduces an audio-first methodology tailored to a morphologically complex, ergative-absolutive language with a four-gender agreement system. Using structured elicitation, crowdsourcing, and translation alignment, the dataset contains approximately 10 hours of curated audio from 42 speakers. Preliminary Whisper-based translation experiments are presented, but the primary contribution is a scalable framework for speech-first corpus design (Saleem et al., 2026).

Hi-SEMFLOW proposes a Lie algebra–based semantic flow framework for span-level labeling in Hindi. It models label refinement as a continuous process using antisymmetric generators. On the HiSlang-4.9k benchmark, it improves span-level F1 by 2–3 absolute points and yields consistent macro-F1 gains, demonstrating effectiveness over discrete structured decoding approaches (Ravikiran et al., 2026).

For Punjabi speech emotion recognition, a dataset covering four emotions (angry, happy, sad, neutral) is introduced. Three models are evaluated: CNN-BiLSTM, ResNet-34, and Wav2Vec 2.0. ResNet-34 achieves the best performance with 96% accuracy in the combined-domain setting (E4), while cross-domain evaluations highlight challenges, especially for neutral emotion classification (Zahra et al., 2026).

The NEGTEG benchmark for Telugu includes five tasks: negation detection, translation, paraphrase detection, sentiment analysis, and polarity flipping. Evaluation shows that multilingual models struggle significantly with Telugu negation across all tasks, particularly due to morphological complexity and linguistic variation (Bairi and Krishnamurthy, 2026).

NeCCo introduces a culturally grounded Nepali commonsense benchmark covering five domains: kinship and social hierarchy; festivals, rituals, and geography; idioms, proverbs, and metaphors; commonsense and daily life; and gastronomy, agriculture, and nature. Evaluation shows models per-

[7] Nepali Romanized social media dataset
[8] Kaggle Dataset
[9] Nepali ASR Dataset
[10] Urdu NewsHeadline Dataset

form better on globally documented knowledge but struggle with culturally dense tasks, exhibiting brittleness, hallucination, and difficulty with implicit reasoning (Shrestha et al., 2026b).

NepaliXlit improves transliteration from Romanized Nepali to Devanagari using 2,943 training word pairs and 736 test pairs. It improves transliteration accuracy by 8% and reduces character error rate by 11%. A dataset of 6,500 comments (with 1,518 annotated instances) shows that transliteration improves sentiment classification, while LLMs outperform encoder models in cross-script scenarios (Patel et al., 2026).

A Tamil tokenization study evaluates WordPiece, SentencePiece, and BBPE using a KaggleTamil news dataset. WordPiece and SentencePiece outperform BBPE in efficiency and classification performance. While BBPE eliminates OOV words, excessive fragmentation harms learning. Increasing vocabulary improves WordPiece and SentencePiece but not BBPE (Sivakumaran et al., 2026).

ILAKKANAM introduces a Tamil linguistic benchmark of 820 questions from Grades 1–13, annotated into five linguistic and factual categories. Evaluation shows Gemini 2.5 performs best overall, while open-source models lag behind. Performance declines with increasing grade level and linguistic complexity, and no strong correlation exists between overall performance and linguistic category identification (Varsha et al., 2026).

Nep-Health-Misinfo is introduced as the first human-verified Nepali health misinformation corpus, built from four existing benchmarks using an MTPE pipeline with native experts. The study finds a strong translation asymmetry, where SOTA models performs much better on factual health content than on deceptive narratives.Translation evaluation shows BLEU drops from 43.21 (factual) to 19.11 (deceptive), with TER reaching 62.42. Few-shot prompting improves Macro-F1 from 0.7188 (zero-shot) to 0.8488 for Qwen2.5-7B, though results remain sensitive to exemplar selection (Maharjan et al., 2026).

A Bengali multitask classification study evaluates LLMs across five domains. Results show chain-of-thought prompting often degrades performance. Gemma-3-4B achieves the most balanced performance across both in- context learning and parameter-efficient fine-tuning settings (Hossain et al., 2026).

For Urdu medical NER, ChatGPT-4o and LLAMA 3.2 are evaluated on 2,057 annotated headlines across five entity types. ChatGPT-4o achieves F1 0.35 (disease), LLAMA 3.2 achieves 0.33, while treatment F1 scores are extremely low (0.011 and 0.036). Overall micro-F1 scores are 0.187 and 0.183, indicating poor performance (Nasim et al., 2026).

The Nwāchā Munā corpus introduces 5.39 hours of Nepal Bhasha speech data. Fine-tuning a Nepali Conformer reduces CER from 52.54% to 17.59%, matching Whisper-Small performance with fewer parameters, demonstrating effectiveness of proximal cross-lingual transfer (Sharma et al., 2026).

The first morphological transducer for Limbu achieves 60% coverage, 88% precision, and 32% recall using a small lexicon derived from field data and Bible translations. It highlights the need for expanded lexical resources and community involvement (Singh and Washington, 2026).

SiPaKosa introduces a corpus of 786K sentences and 9.25M words from 16 historical documents and canonical texts. Evaluation across 10 models shows perplexity ranges from 1.09 to 189.67, with proprietary models outperforming open-source ones by 3–6 times (Gururssinghe and Jayatilleke, 2026).

A reward-guided fine-tuning approach for Nepali ASR uses 2,000 human-rated samples to train a classifier with 81% accuracy to filter bad samples. Filtering twice and retraining ASR on 40,000 clip training subset drawn from a 68.4 hour corpus improves WER from 5.60% to 4.89% and CER from 5.10% to 4.52%, yielding 11–13% relative gains over baseline (Pandit et al., 2026).

A study on LoRA for Hindi-English code-mixing through spectral analysis of mBERT and MuRIL shows pre-trained attention ranks of 437–441, while LoRA updates have ranks 2.1–5.9, achieving $136\times$ compression. CKA scores show alignment of 0.279 (Hindi) vs. 0.093 (English), with statistical significance (Wilcoxon $p < 10^{-19}$), supporting low-dimensional subspace hypotheses (Vishwakarma and Kumar, 2026).

DR-RAG introduces dual-representation retrieval (text chunks + QA pairs) for Urdu. It improves Urdu METEOR by 38×, ROUGE-1 by 140%, and reduces latency by 43%. LLM-as-judge scores improve from 1.93 to 3.03 (faithfulness) and 2.21 to 2.99 (overall quality), demonstrating effectiveness of representation alignment between queries and indexed content.(Ahmad et al., 2026).

5. Key Takeaways

1. **Severe Resource Scarcity Remains the Central Bottleneck.** Many works highlight the lack of high-quality datasets across South Asian languages such as Bengali, Tamil, Nepali, Urdu, Punjabi, and Burushaski, which continues to hinder model development and evaluation (Haque et al., 2026; Nerujan and Sarveswaran, 2026; Saleem et al., 2026; Zahra et al., 2026; Nasim et al., 2026; Shrestha et al., 2026b).

2. **Dataset Quality and Linguistic Curation Are**

Critical. Carefully designed and linguistically grounded datasets significantly improve reliability, interpretability, and evaluation of models (Haque et al., 2026; Shrestha et al., 2026b; Varsha et al., 2026; Maharjan et al., 2026; Bairi and Krishnamurthy, 2026).

3. **Multimodal Understanding is Essential but Challenging.** Meme-based communication requires integrating textual and visual signals, and unimodal approaches fail to capture full semantics in low-resource settings (Bansal et al., 2026; Orny et al., 2026; Regmi et al., 2026; Shrestha et al., 2026a).

4. **Fusion Strategies Strongly Influence Performance.** The effectiveness of early fusion, late fusion, cross-modal attention, stacking, and voting strategies varies depending on task type and modality interaction (Acharya et al., 2026; Fondekar et al., 2026; Wagle et al., 2026; Muntaha et al., 2026).

5. **Low-Resource Settings Expose Weaknesses of Current Models.** State-of-the-art models struggle with linguistic phenomena such as negation, cultural reasoning, and domain-specific tasks, indicating shallow understanding in low-resource contexts (Bairi and Krishnamurthy, 2026; Shrestha et al., 2026b; Nasim et al., 2026; Hossain et al., 2026).

6. **Morphological Complexity and Tokenization Are Major Issues.** Rich morphology and subword fragmentation significantly affect representation learning and downstream performance (Nerujan and Sarveswaran, 2026; Sivakumaran et al., 2026; Ahmad et al., 2026).

7. **Code-Mixing and Script Variation Add Significant Complexity.** Mixed scripts and informal language usage create challenges for multilingual models, though transliteration and cross-lingual strategies provide improvements (Patel et al., 2026; Fondekar et al., 2026; Vishwakarma and Kumar, 2026).

8. **Speech and Oral Languages Require Different Paradigms.** For languages with limited written resources, speech-first approaches and audio-centric datasets are more suitable than text-based pipelines (Saleem et al., 2026; Sharma et al., 2026).

9. **Data-Centric and Lightweight Methods Are Effective Alternatives.** Improvements can be achieved through data filtering, human-in-the-loop approaches, proximal transfer, and parameter-efficient methods (Pandit et al., 2026; Sharma et al., 2026; Hossain et al., 2026).

10. **Cultural and Contextual Knowledge Is Crucial for True Language Understanding.** Models trained on global datasets fail to capture culturally embedded knowledge, leading to poor performance in culturally grounded tasks (Shrestha et al., 2026b).

6. Shared Task Description

In addition to the main track, the workshop hosted a shared task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes, focusing on monolingual Nepali memes written in the Devanagari script. The task comprised two subtasks: (1) hate speech detection and (2) sentiment analysis, both requiring joint reasoning over textual and visual modalities. The shared task attracted strong participation from the community, with 65 registered participants leading to 23 teams submitting systems for hate detection and 43 registered participants with 13 teams submitting systems for sentiment analysis. The diversity of submissions reflects a wide range of approaches, including multimodal fusion architectures, caption-based methods using vision-language models, and ensemble techniques. Further details on the dataset, task design, and evaluation protocol are provided in (Thapa et al., 2026).

The submitted systems demonstrated that multimodal approaches are generally more effective than unimodal baselines in low-resource settings. The top-performing system achieved macro-F1 scores of 80.52% for hate speech detection and 68.81% for sentiment analysis using a late-fusion hybrid architecture combining strong multilingual text encoders with vision models and employing discriminative learning rates. Overall, the results indicate that hate detection is relatively more tractable than sentiment analysis, which requires capturing finer-grained semantic and cultural nuances. Despite promising performance, the task also highlights persistent challenges, including the limited availability of Devanagari-centric pretrained models and the difficulty of modelling culturally grounded multimodal content, pointing to important directions for future research.

7. Conclusion and Future Directions

The second edition of the CHiPSAL workshop demonstrates the growing interest and momentum in research on South Asian language processing. The workshop received a 57 of submissions, covering a wide range of languages, tasks, and modalities. Notably, the accepted papers span both widely spoken and underrepresented languages, including Bengali, Hindi, Nepali, Pali, Punjabi, Sinhala,

Tamil, Telugu, and Urdu, as well as extremely low-resource and endangered languages such as Burushaski, Limbu, and Nepal Bhasha (Newari). Several contributions focused on resource-intensive efforts, as outlined in Section 3, including the development of datasets and tools for these languages across both text and speech domains.

These contributions represent significant progress when considered in the context of limited funding, infrastructural constraints, and organisational challenges that characterise research in this region. These efforts are particularly valuable, as they lay the groundwork for future research and resource development, contributing to a more inclusive and representative NLP landscape.

The strong response to the workshop, with 57 submissions for a half-day event, highlights the importance and timeliness of initiatives such as CHiPSAL. At the same time, it is important to acknowledge that participation in major international venues remains challenging for many researchers in the region due to financial, institutional, and organisational constraints. Addressing these barriers is crucial for ensuring broader and more equitable participation in the global research community.

In light of this, we plan to expand CHiPSAL through regional initiatives to better support local research communities. The first such event is planned to be held in Sri Lanka on 31 July 2026. We invite researchers and institutions across South Asia to collaborate in organising similar satellite events, with the aim of strengthening regional networks and fostering sustained and inclusive research engagement in South Asian language technologies.

8. Acknowledgments

We would like to thank Ahrane Mahaganapathy, Ajintha Sivakulasingam, and Nishanthini Kanthakumar for their valuable assistance as volunteers in organising the workshop.

We are grateful to all members of the programme committee for their timely reviews despite the short timeframe; their contributions have significantly improved the quality of the workshop.

We also thank the LREC workshop chairs for accepting our proposal and for their support in facilitating the successful organisation of the workshop.

9. Bibliographical References

Ashish Acharya, Anish Khatiwada, Rohit Khadka, and Pragya Aryal. 2026. TeamHerald@CHIPSAL 2026: Hate Speech Detection and Sentiment Analysis of Nepali Memes using Transformer-based Architectures and Ensemble Learning. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Saad Ahmad, Muhammad Hammad, Muhammad Zeeshan, Faizad Ullah, and Asim Karim. 2026. DR-RAG: Addressing Retrieval Misalignment in Low-Resource Urdu Question Answering. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Vennela Bairi and Parameswari Krishnamurthy. 2026. Lost the Negation or Lost in Negation. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Vinayak Bansal, Deepawali Sharma, Aakash Singh, and Vivek Kumar Singh. 2026. EthosAI@CHiPSAL2026: Hate and Sentiment Understanding in Low-Resource Memes using a Multimodal Approach. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Ashweta A. Fondekar, Milind M. Shivolkar, and Jyoti D. Pawar. 2026. Unigoa@CHiPSAL 2026: Early vs Late Fusion for Multimodal Hate and Sentiment Detection in Nepali Memes. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Ranidu Gurursinghe and Nevidu Jayatilleke. 2026. SiPaKosa: A Comprehensive Corpus of Canonical and Classical Buddhist Texts in Sinhala and Pali. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Farah Binta Haque, Md Yasin, Shishir Saha, Md Shoaib Akhter Rafi, and Farig Sadeque. 2026. BNLI: A Linguistically-Refined Bengali Dataset for Natural Language Inference. In *Proceedings of the Second Workshop on Challenges*

in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Hans Henrich Hock and Elena Bashir, editors. 2016. *The Languages and Linguistics of South Asia*. De Gruyter Mouton, Berlin, Boston.

Md. Sajjad Hossain, Kawsar Ahmed, Suny Md Ashraf Khan, and Mohammed Moshiul Hoque. 2026. Exploring Large Language Models for Multitask Learning in Bengali Text Classification. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Sujal Maharjan, Astha Shrestha, Laxmi Thapa, Sweta Poudel, Shuvam Shiwakoti, Rabin Thapa, Kritesh Rauniyar, and Surendrabikram Thapa. 2026. Improving Public Health Safety in Low-Resource Languages Using a Human-Verified Health Misinformation Corpus and Large Language Models. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Sidratul Muntaha, Sabila Anzum, Arpita Mallik, and Hasan Murad. 2026. NeuralNoodles@CHiPSAL 2026: Late-Fusion Multimodal Stacking for Nepali Meme Sentiment Classification. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Bushra Nasim, Kinza Latif, Muhammad Zohair, Muhammad Hassan Asif, and Zarmeen Nasim. 2026. Evaluating Large Language Models for Medical Named Entity Recognition in Urdu: A Benchmark Study. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Sathasivam Nerujan and Kengatharaiyer Sarveswaran. 2026. A Feature-Fusion Ensemble Approach for Tamil Hate Speech Detection. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Noore Tamanna Orny, Joyeta Barua Moni, Md. Abtahee Kabir, and Hasan Murad. 2026. Team Oryu@CHiPSAL 2026: Integrating Text and Vision Transformers for Multimodal Hate Speech Detection in Memes. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Aadarsh Pandit, Yudhin Khanal, Ishan Pandey, Kushal Kunwar, and Sunil Regmi. 2026. Reward-Guided Fine-Tuning of Whisper for Low-Resource Nepali Speech Recognition. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Suraj Patel, Kashish Kumari Dhami, Norden Sherpa, and Supriya Khadka. 2026. From Romanized to Devanagari: Enhancing Nepali Sentiment Analysis with NepaliXlit. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Manikandan Ravikiran, Tanmay Tiwari, Vibhu Gupta, and Rohit Saluja. 2026. Hi-SEMFLOW: Lie Algebra–Based Semantic Flow for Span-Level Informal Language Identification in Hindi. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Sunil Regmi, Bipesh Subedi, Saugat Singh, and Suman Shrestha. 2026. linus@CHiPSAL 2026: Multimodal Hate Speech and Sentiment Detection in Low-Resource Memes using Late-Fusion Hybrid Architecture. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Tauqeer Saleem, Abdul Samad, Azkaa Nisar, Fatima Faisal, Adina Adnan Mansoor, and Mahrukh Yousuf. 2026. Development of Burushaski Speech – English Text Translation Dataset. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Kengatharaiyer Sarveswaran, Gihan Dias, and Miriam Butt. 2021. Thamizhi morph: A morpho-

logical parser for the tamil language. *Machine Translation*, 35(1):37–70.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Sana Shams, Ashwini Vaidya, and Bal Krishna Bal. 2025. A brief overview of the first workshop on challenges in processing South Asian languages (CHiPSAL). In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 1–8, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Rishikesh Kumar Sharma, Safal Narshing Shrestha, Jenny Poudel, Rupak Tiwari, Arju Shrestha, Rupak Raj Ghimire, and Bal Krishna Bal. 2026. Nwāchā Munā: A Devanagari Speech Corpus and Proximal Transfer Benchmark for Nepal Bhasha ASR. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Sandesh Shrestha, Bikram K.C., Akshyat Shah, Ashish Acharya, and Rabin Thapa. 2026a. Multi-Modal-Minds@CHiPSAL 2026: A Comparative Study of Textual, Visual and Multimodal Architecture for Nepali Meme Moderation. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Sanket Shrestha, Raunak Regmi, Sadikshya Ghimire, Satyam Rana, and Supriya Khadka. 2026b. NeCCo: Nepali Cultural Commonsense Benchmark for Large Language Model Evaluation. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Avyaya Singh and Jonathan North Washington. 2026. A Morphological Transducer for the Limbu Language. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Gokulan Sivakumaran, Randil Pushpananda, and ERAD Bandara. 2026. Comparative Analysis of Tokenizers in Tamil Text Classification in Low Resource Settings. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Jeyarajalingam Varsha, Menan Velayuthan, Sumirtha Karunakaran, Rasan Nivethiga, and Kengatharaiyer Sarveswaran. 2026. Evaluating Linguistic Knowledge of LLMs in Tamil: The *ILAKKANAM* Benchmark. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Shashank Vishwakarma and Rakesh Kumar. 2026. Why Does Low-Rank Adaptation Work for Hindi-English Code-Mixing? A Geometric Analysis. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Samir Wagle, Reewaj Khanal, and Abiral Adhikari. 2026. MEME-Fusion@CHiPSAL 2026: Multimodal Ablation Study of Hate Detection and Sentiment Analysis on Nepali Memes. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Fatima Tu Zahra, Kulsoom Asim, Sandesh Kumar, and Abdul Samad. 2026. Cross-Domain Evaluation of Transformer-Based Models for Punjabi Speech Emotion Recognition. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) @ LREC 2026*, Palma, Mallorca (Spain). European Language Resources Association (ELRA).

Development of Burushaski Speech – English Text Translation Dataset

Tauqeer Saleem, Dr Abdul Samad, Azkaa Nisar, Fatima Faisal, Adina Adnan Mansoor, Mahrukh Yousuf

Habib University
Karachi, Pakistan
{tauqeer.saleem, abdul.samad}@sse.habib.edu.pk, {an08017, ff08435, am08349, my08055}@st.habib.edu.pk

Abstract

Burushaski is a language isolate spoken in northern Pakistan with a predominantly oral tradition, limited standardized orthography, and virtually no existing speech technology infrastructure. These characteristics make conventional text-centric NLP pipelines unsuitable and position speech data collection as the primary scientific challenge. This paper introduces an audio-first, linguistically informed methodology for developing a Burushaski–English speech translation resource. Rather than prioritizing model architecture, we focus on principled corpus design tailored to the language's morphological complexity, ergative-absolutive alignment, and four-gender agreement system. The dataset combines structured elicitation targeting high-frequency and morphologically diverse constructions, functional and formulaic speech, and oral narratives that capture discourse-level phenomena.

We describe the design of a custom data collection application, community-embedded crowdsourcing strategy, and translation-aligned workflow for generating parallel speech–English data. The resulting pilot corpus comprises approximately 10 hours of curated audio from 42 speakers across controlled and naturalistic settings.

While we present the results of preliminary translation experiments using Whisper, the primary contribution of this work is methodological: a scalable framework for speech-first corpus development in morphologically rich, under-resourced, and predominantly oral languages. We argue that for languages lacking stable orthography and large textual corpora, data design, not model selection constitutes the central research problem

Keywords: Burushaski, Translation, Textless NLP

1. Introduction

Recent advances in self-supervised speech modeling have significantly reduced the dependence of automatic speech recognition (ASR) and speech translation systems on large manually transcribed corpora. Models such as wav2vec 2.0 and XLS-R demonstrate that robust acoustic representations can be learned from raw audio at scale. However, these advances do not eliminate a more fundamental bottleneck: the absence of well-designed, representative speech datasets for languages that lack standardized orthography, large textual corpora, and existing speech technology infrastructure. For predominantly oral and morphologically complex languages, corpus design remains the central challenge.

Burushaski, spoken in the Hunza, Nagar, and Yasin valleys of northern Pakistan, exemplifies this setting. As a language isolate with a primarily oral tradition and limited orthographic standardization, Burushaski has minimal digitized textual resources and virtually no publicly available speech corpora. With an estimated 90,000 -120,000 speakers across three mutually intelligible but structurally distinct dialects - Hunza, Nagar, and Yasin. While multilingual self-supervised models may offer cross-lingual acoustic transfer, effective downstream adaptation still requires carefully structured language-specific data. In addition, Burushaski exhibits typological properties, such as including ergative-absolutive alignment, a four-gender noun class system, and rich verbal morphology that increase data sparsity under naive collection strategies. Opportunistic or read-speech corpora risk underrepresenting inflectional paradigms and discourse patterns essential for generalizable modeling.

In such contexts, the primary scientific question shifts from model architecture to dataset construction. How should speech data be elicited, structured, and validated to ensure morphological coverage, phonetic diversity, and functional relevance for downstream translation tasks? How can parallel speech–translation resources be created when textual norms are unstable and literacy practices vary across speakers? Addressing these questions requires integrating linguistic field methodology, speech technology constraints, and community-embedded collection practices.

This paper presents an audio-first, linguistically informed framework for developing a Burushaski–English speech translation resource. Rather than prioritizing architectural novelty, we focus on principled corpus design tailored to the structural properties of the language and the practical realities of its speaker community. Our approach combines (i) structured elicitation targeting high-frequency vocabulary and systematic coverage of

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 9–17
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

tense–aspect–mood and agreement paradigms, (ii) collection of functional and formulaic language relevant to real-world applications, and (iii) oral narratives and folktales capturing discourse-level phenomena and natural variation. Data collection is facilitated through a custom mobile application designed to standardize prompts while enabling scalable community participation.

The resulting pilot corpus comprises approximately ten hours of curated Burushaski speech paired with English translations, collected from 42 speakers across controlled and semi-naturalistic settings.

We report corpus design principles, speaker distribution, and linguistic coverage characteristics, and we outline baseline experiments using Whisper for translation. These modeling experiments are exploratory; the principal contribution of this work lies in establishing a replicable methodology for speech-first corpus development in morphologically rich, low-resource languages with limited textual infrastructure. Concretely, this paper makes the following contributions: (1) the first parallel Burushaski–English speech corpus, comprising approximately 10 hours of audio from 42 speakers across controlled and semi-naturalistic settings; (2) a custom mobile data collection application designed for low-infrastructure, low-literacy field settings; and (3) baseline translation experiments with Whisper establishing a reference point for future work.

2. Related Work

2.1 Speech Technology in Low-Resource and Predominantly Oral Languages

Recent advances in self-supervised learning have reshaped low-resource speech technology. Models such as Whisper, wav2vec 2.0 (Baevski et al,. 2020) and XLS-R (Conneau et al., 2021) demonstrate that acoustic representations can be learned from large volumes of unlabeled speech and subsequently fine-tuned for downstream tasks, including automatic speech recognition (ASR) and speech translation. Multilingual pretraining has further enabled cross-lingual transfer, allowing models trained on high-resource languages to adapt to lower-resource settings with comparatively limited labeled data.

However, the effectiveness of such transfer depends on the availability of at least modest quantities of representative speech data for fine-tuning. For languages with unstable orthography, limited digitized text, and little existing speech infrastructure, the primary bottleneck is not model architecture but the absence of systematically designed corpora. Predominantly oral languages introduce additional challenges: read-speech datasets may not reflect natural prosody or discourse patterns, and inconsistent orthographic practices complicate alignment and evaluation. These constraints shift the focus from model selection to data design and collection methodology.

2.2 Corpus Design and Linguistic Coverage in Morphologically Rich Languages

Low-resource speech performance is strongly influenced by the linguistic coverage of the underlying dataset. In morphologically rich languages (MRLs), surface form variation increases data sparsity: a single lexical root may appear across numerous inflectional forms encoding case, agreement, tense, aspect, or mood. Opportunistic corpus collection—such as harvesting spontaneous or read text without structural planning—often underrepresents critical morphological paradigms, limiting generalization.

Prior work highlights different strategies to address this issue. The Mboshi corpus described by Godard et al. (2017) integrated language documentation practices with computational alignment, collecting speech paired with French translations using mobile elicitation tools. Their approach prioritized natural speech while maintaining parallel structure suitable for downstream modeling. In contrast, phonetically balanced corpus construction efforts (e.g., Gupta et al., 2016) emphasize systematic coverage of phonemes and triphones to improve acoustic robustness.

These approaches demonstrate that linguistic planning—whether targeting phonetic coverage or grammatical paradigms—is essential in low-resource settings. Yet relatively little work formalizes elicitation strategies that simultaneously target (i) morphological coverage, (ii) high-frequency functional vocabulary, and (iii) discourse-level naturalness within a unified speech-translation framework. For languages with complex alignment systems and agreement structures, corpus design must explicitly address paradigm diversity rather than rely solely on frequency-based sampling.

2.3 Speech-to-Text Translation in Low-Resource Setting

End-to-end speech-to-text translation (ST) systems have gained prominence as alternatives to cascaded ASR-plus-machine-translation pipelines. Multilingual encoders combined with Transformer decoders have demonstrated competitive performance in several low-resource scenarios when supported by parallel speech–text data. Multilingual pretraining has been shown to reduce supervision requirements, particularly when typologically related languages are included in the training distribution.

However, most low-resource ST studies assume the existence of standardized orthography and

curated parallel corpora. For predominantly oral languages, creating aligned speech–translation data is itself a nontrivial task. Orthographic instability complicates transcription consistency, while limited literacy may constrain text-based data collection workflows. As a result, methodologies for constructing parallel speech–translation datasets remain underdeveloped for languages lacking established textual norms.

In such cases, translation-aligned elicitation—where speech is collected directly with corresponding translations—offers an alternative pathway that bypasses dependence on large native-language text corpora. The effectiveness of this approach depends on careful prompt design and validation to ensure linguistic diversity and functional relevance.

2.4 Community-Embedded Data Collection and Crowdsourcing

Crowdsourcing has become a common strategy for scaling speech data collection. However, studies have shown that conventional online platforms systematically underrepresent speakers of low-resource and rural languages due to infrastructure, literacy, and accessibility constraints. Field-based and community-embedded approaches can mitigate these limitations by tailoring tools and workflows to local contexts.

Mobile data collection applications have been successfully used in documentation-oriented corpora to standardize prompts while preserving naturalistic speech production. Translation-based elicitation further enables the creation of parallel corpora without requiring extensive written materials in the target language. Nevertheless, ensuring speaker diversity, gender balance, and dialectal variation remains an ongoing challenge, particularly in small pilot datasets.

2.5 Positioning of the Present Work

Existing research demonstrates the value of self-supervised speech encoders, linguistically informed corpus design, and translation-aligned data collection. Yet there remains a methodological gap in integrating these strands for morphologically complex, predominantly oral languages that lack stable orthography and prior speech infrastructure.

The present study addresses this gap by formalizing an audio-first corpus design framework for Burushaski that:

1. Systematically targets morphological paradigms (ergative-absolutive alignment, agreement, tense–aspect–mood distinctions);
2. Incorporates high-frequency functional and formulaic language relevant to practical translation scenarios;
3. Integrates oral narratives to capture discourse-level phenomena and natural variation; and
4. Embeds data collection within a structured mobile application workflow enabling scalable community participation.

Rather than proposing a novel model architecture, this work treats dataset construction as the primary research contribution and evaluates baseline multilingual encoders within that framework. By foregrounding corpus design as a central methodological problem, the study contributes a replicable approach for developing speech-translation resources in similarly under-documented and typologically complex languages.

3. Dataset Collection

Developing speech resources for a predominantly oral and morphologically rich language requires deliberate corpus design rather than opportunistic data gathering. The dataset collection strategy for Burushaski was therefore structured around three principles: (i) systematic linguistic coverage, (ii) functional relevance for translation tasks, and (iii) scalable community participation through controlled prompting.

3.1 Corpus Design Framework

The dataset was constructed using a category-based elicitation protocol defined in a master spreadsheet of English–Burushaski sentence pairs. Our approach systematically targets Burushaski's morphological complexity through stimulus design informed by standardized linguistic questionnaires (Lahiri 2018, Max Planck Field Manuals), generating parallel text-audio alignments essential for textless pipeline training and evaluation. Sentences were grouped into linguistically motivated categories, including:

- Simple and complex sentence structures capturing the basic syntactic organization of the language
  (e.g., "The monkey is on the tree.")
- Case markers that target ergative, absolutive, and other case distinctions
  (e.g., "She (right here) hit the bear.")
- Tense, aspect, and mood contrasts that elicit inflectional variation across temporal and aspectual paradigms
  (e.g., "I am walking to the field now.")
- Negative constructions capturing negation morphology across sentence types
  (e.g., "There is no water in the river.")
- Reflexive constructions targeting self-referential agreement patterns
  (e.g., "We saw ourselves in the mirror.")
- Serial and converb verb constructions (ECV) capturing multi-predicate and chained verb structures
  (e.g., "She has gone outside the house.")

- Noun classifiers eliciting Burushaski's gender-class agreement system *(e.g., "People have gathered here.")*
- Interrogatives covering both content and polar question formation *(e.g., "Where are you going?")*
- Pronouns targeting person, number, and gender paradigms *(e.g., "You are late.")*
- Valency alternations contrasting transitive and intransitive verb frames *(e.g., "He opened the door." vs. "The door opened.")*
- Causative constructions eliciting morphological and periphrastic causation *(e.g., "Mother made me go outside.")*
- Possession constructions covering attributive and predicative possession *(e.g., "I have a book.")*
- Adpositions and postpositions capturing spatial and relational marking *(e.g., "She is in the room.")*
- Passive constructions targeting voice alternations and agent demotion *(e.g., "The door was opened by the boy.")*
- Imperative constructions eliciting directive speech acts *(e.g., "Sit down.")*
- Naturalistic narrative data capturing discourse-level phenomena and spontaneous speech variation *(e.g., "What is the climate like where you live?")*

Rather than collecting isolated lexical items, each category was designed to trigger controlled variation within grammatical paradigms. For example, minimal pairs such as:

- "She (right here) saw a man."
- "She (not here) saw a man."

These two sentences are identical in meaning, The only difference is whether the person, "she" is present near the speaker or not. In Burushaski, this distinction is grammatically marked, the verb and pronoun forms change depending on the location of the subject relative to the speaker. This small variation can reliably elicit and isolate the specific morphological forms Burushaski uses to encode deictic contrast. This controlled design ensures that these distinctions are captured across multiple sentence contexts rather than appearing only once in the corpus, reducing sparsity in the training data.

Similarly, parallel prompts varying subject gender or participant role target inflectional alternations without altering lexical content. This structured variation reduces morphological sparsity by ensuring that inflectional paradigms are represented across multiple lexical roots.

The design deliberately balances frequency-based vocabulary with morphosyntactic coverage. High-frequency functional language (e.g., kinship terms, classroom interactions, temporal expressions) ensures relevance for real-world translation scenarios, while targeted paradigm elicitation supports downstream modeling robustness. For further robustness, we have also used Common Voice's dataset for Burushaski[1], this dataset had only audios and transcript without English translation, so we asked volunteers to translate it for us. The source dialect for this dataset is that of Hunza.

3.2 Translation-Aligned Elicitation

Given the limited availability of standardized written Burushaski, the collection process followed a translation-aligned workflow. English prompts were used as structured stimuli which target frequency based everyday vocabulary across general contexts (Kilgarriff et al., 2014; Finlayson et al., 2024), and native speakers produced corresponding Burushaski speech recordings. This approach ensures parallel speech–English data without requiring large native-language textual corpora.

To increase ecological validity, prompts were written in simple, contextually plausible English rather than artificial grammatical templates. Where possible, multiple speakers recorded the same prompt to capture phonetic and prosodic variation while preserving semantic alignment.

In addition to structured elicitation, selected folktale excerpts and narrative prompts were included to capture discourse-level phenomena, natural prosody, and code-switching behavior. This hybrid design combines controlled grammatical targeting with naturalistic speech segments.

To ensure the validity of the translations, we assigned human validators to ensure quality and accuracy. Each Burushaski recording was reviewed by 3 independent native-speakers, who verified that the spoken output accurately and naturally conveyed the meaning of the original English prompt. This validation pipeline was applied consistently across both the structured elicitation material and the narrative/folktale segments, ensuring dataset-wide translation fidelity.

3.3 Mobile Application Infrastructure

To standardize collection and enable distributed participation, we developed a mobile recording

[1] Bsk Common Voice Dataset: https://datacollective.mozillafoundation.org/datasets/cmj8u3owx003xnxxb3f7n3m1l

application that presents prompts sequentially and records corresponding speech in similar fashion to Godard et al. (2017). Each prompt appears with:

- An English stimulus sentence
- Its Burushaski equivalent (where applicable)
- A recording interface
- Submission and validation feedback – volunteers and research assistants were part of the validation process to ensure that the given Burushaski equivalent is correct and that the community is involved at each step of the data collection process.

The application enforces uniform recording procedures, reducing variability introduced by heterogeneous recording setups. Each submission is logged with metadata including speaker identity (anonymized ID), sentence ID, and timestamp. Participants ranged in age from approximately 18 to 60 years, with the majority recruited from Hunza and Nagar. Quality assurance was ensured within the community workflow at two distinct stages rather than relying solely on post-hoc expert review.

Stage 1 – Prompt Validation (pre-recording) Before any prompt was made available to speakers, research assistants and community volunteers confirmed the Burushaski equivalent displayed alongside each English sentence was accurate and natural. Items flagged as incorrect or unidiomatic were withdrawn, corrected, and re-reviewed before re-release. This ensured no speaker was provided with erroneous or unnatural Burushaski text.

Stage 2 – Post-recording audio review. Submitted recordings were reviewed whether (a) the spoken utterance conveyed the meaning of the English prompt (b) was grammatically natural and fluent, and (c) audio quality free for noise, clipping or truncation.

Speaker representation across the three primary dialect regions - Hunza, Nagar, and Yasin was uneven in this initial phase, with Yasin speakers significantly underrepresented, broadening dialectal coverage is a priority for future collection rounds.

This structured interface serves two purposes:

1. Ensuring consistent prompt delivery across participants.
- Enabling scalable, community-driven data collection without requiring advanced technical literacy.

3.4 Ethics, Consent and Data Governance

All participants provided informed consent prior to recording. Consent and demographic information were collected digitally via a structured Google Form. The form captured participant age, gender, dialect region alongside explicit consent for the use of recordings in research. Participation was entirely voluntary and participants were informed they could withdraw at any time. Speaker identities are retained internally only as anonymized IDs; no personally identifiable information is included in the released dataset. The Google forms responses are stored securely and separately from the audio data, accessible only to the research team.

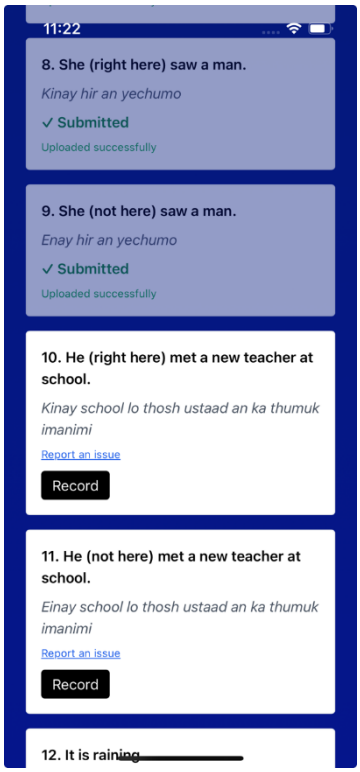


Figure 1: Screenshot of the mobile application for data collection

4. Translation Model – Whisper

Whisper uses a Transformer-based encoder to process log-Mel spectrogram inputs, and a Transformer decoder to autoregressively generate text tokens. The model supports both speech-to-source transcription and direct speech-to-English translation. In this work, we used direct speech-to-English translation mode, bypassing intermediate Burushaski transcription.

We fine-tuned three variants of Whisper: Whisper-Small (~244M parameters), Whisper-Medium (~769M parameters) and Whisper-Large-v2 (~1.55B parameters). All experiments used the same dataset of 7,115 examples drawn from a

combined pool of Mozilla Common Voice Burushaski recordings and app-collected data, which were merged into a single corpus prior to splitting. The dataset was partitioned using a fixed random seed (seed=42) into a training set of 6,403 examples (90%) and a validation set of 712 examples (10%), with a held-out test set of 1,779 samples used exclusively for final evaluation.

Pre processing and data preparation was the same for all three models. Audio was resampled at 16kHz and converted into 80-channel log-mel spectrograms using Whisper's default feature extractor. English target translations were lowercased and stripped of punctuation and excess whitespace before tokenization. Audio durations ranged from 0.5s to 14.3s (avg: 3.2s), and English reference lengths ranged from 1 to 41 words (avg: 5.4 words).

The training parameters for Whisper-Small and Whisper-Medium were the same, with only slight modifications made for Whisper-Large-v2-v2 because of GPU memory limitations. Every model was trained using a learning rate of 1e-5, a warmup of 200 steps, mixed precision training (fp16), a batch size of 8 per device with two gradient accumulation steps, and gradient checkpointing enabled.

Fine-tuning was performed using paired speech–English translation targets. No intermediate ASR step or external machine translation system was used; the model was trained end-to-end for direct translation. Given (a) lack of standardized Burushaski orthography, (b) limited transcription resources, (c) desire to avoid cascading ASR error propagation, direct speech-to-English translation was selected over ASR+MT pipelines. This approach removes dependence on consistent native-language text while aligning with Whisper's pretraining objective. The Mozilla Common Voice recordings and app-collected data were not treated as separate streams during training; both sources were merged into a single dataset and shuffled before the train/test/validation split was applied, ensuring that both data sources were represented across all splits.

5. Results

5.1 Dataset Summary

We have successfully started the process of gathering data that is representative of the language, the process will continue and deliver better quality data in the coming years. More sentence categories will be added, and more literature will be added. Table 1 summarises the pilot corpus collected to date. The 5,095 application recordings from 42 speakers cover 511 unique prompt sentences, yielding an average of approximately 10 recordings per sentence, sufficient for capturing phonetic and prosodic variation across speakers for most prompts. The gender distribution is notably skewed, with female speakers contributing 3,642 recordings (71%) compared to 1,453 from male speakers (29%), this imbalance reflects recruitment constraints in the pilot phase and is a known limitation of the current dataset. The Mozilla Common Voice component (~6 hours) contributes naturalistic speech, though its dialectal distribution is uncontrolled. Combined, the ~10-hour corpus represents the largest known Burushaski–English speech resource, though it remains small relative to the data requirements of standard neural translation systems

Public release is planned following completion of data validation and speaker consent review.

Total Number of Recordings (App)	5095
Total Duration (App)	~4 hours
Participants	42
Unique Sentences	511
Average Utterance Length	~3 seconds
Average Sentences per Category	~30
Dialect Distribution	Hunza: 4,867 (95.5%) Yasin: 228 (4.5%)
Gender distribution of number of recordings	Male: 1,443 (28%) Female: 3,642 (72%)
Average time per participant	121.3 seconds (~2 minutes)
Mozilla Dataset Translated	~6 hours
Total Data in hours	**~10 hours**

Table 1: Summary of data collected

5.2 Translation

Evaluation was conducted on 244 held-out test samples using standard speech translation and transcription metrics.

All three Whisper variants were evaluated on the same held-out test set of 1,779 samples. Decoding was performed using default greedy generation with a maximum output length of 225 tokens. Before metric computation, both predictions and references were normalized by lowercasing and removing punctuation and excess whitespace. BLEU and chrF++ were computed on normalized text; BERTScore F1 was computed on the original unnormalized text using bert-base-multilingual-cased; WER and CER were computed on normalized text using the jiwer library.

Metric	Whisper-Small	Whisper-Medium	Whisper-Large-v2
WER	34.33%	25.71%	24.54%
CER	25.94%	20.90%	19.64%
Word Accuracy	65.67%	74.29%	75.46%
BLEU	65.42	69.94	70.90
chrF++	71.19	77.15	78.24
BERTScore F1	0.9195	0.9375	0.9402

Table 2: *Evaluation results across Whisper model variants on the held-out test set (n=1,779). Lower is better for WER and CER, higher is better for all other metrics.*

The model scale consistently improves the results for all three variants. With a BLEU of 70.90, WER of 24.54%, and BERTScore F1 of 0.9402, Whisper-Large-v2-v2 performs the best. It is closely followed by Whisper-Medium (BLEU 69.94, WER 25.71%) and Whisper-Small (BLEU 65.42, WER 34.33%). Predictions are both lexically close to the references and semantically coherent, which is a significant improvement over the pilot baseline, as evidenced by the comparatively small difference between BERTScore and BLEU across all models.

Furthermore, we did a qualitative error analysis with the help of native Burushaski speakers and realized the reason why there is a discrepancy between BERTScore and BLEU.

1. We found that in several cases Whisper was attempting to get semantic approximation but fails to preserve culturally specific lexical items. For example:
   Reference: “It is the story of Pegasus and a jinn”
   Prediction: “It is the story of the Wind Horse and the Ghost”
2. In other cases, model demonstrated semantic compression, where meaning is preserved but structure differs—contributing to high BERTScore but elevated WER. For example:
   a. Reference: "The wind has been blowing since yesterday."
   b. Prediction: "It has been windy since yesterday."
3. There were several cases where named entities are replaced with semantically plausible but incorrect phrases, reflecting uncertainty in acoustic mapping. For example:
   a. Reference: "I am standing between Ammy and Mary."
   b. Prediction: "I am standing between the two rivers."

The results reflect the tension between large-scale multilingual supervision and language-specific adaptation:

1. Whisper’s pretraining enables semantically coherent English output.
2. However, absence of Burushaski exposure during original training leads to lexical instability and hallucination.

- Fine-tuning on ~10 hours give a strong translation performance, with the remaining errors concentrated in culturally specific expressions and named entities rather than general fluency.

These findings reinforce the central thesis of this study: strong performance is achievable with carefully structured parallel data, and remaining limitations point to the need for broader cultural and lexical coverage rather than architectural changes.

6. Future Work

The present work establishes a methodological foundation and baseline system for Burushaski speech-to-text translation; several directions for improvement are planned for future iterations.

On the data side, the corpus will be expanded substantially through continued community-embedded collection. Additional sentence categories will be added to the elicitation protocol, with particular attention to underrepresented morphological paradigms and domain-specific vocabulary. Natural conversational data and extended oral narratives , including informal dialogues and unscripted community exchanges, will be collected to complement the currently elicitation-heavy corpus. This will address the prosodic and discourse-level gaps inherent to prompted elicitation, following the approach of Godard et al. (2017), who demonstrated that integrating spontaneous and controlled speech yields more generalizable ASR performance. Speaker diversity will also be broadened to achieve better coverage across age groups, genders, and the three primary dialect regions of Hunza, Nagar, and Yasin. The translation of additional Burushaski literary and community texts will provide supplementary parallel data outside of the elicitation framework.

On the modeling side, several more advanced architectures will be explored as the dataset grows. The prosody-aware Generative Spoken Language Model (pGSLM) of Kharitonov et al. (2021) extends the GSLM framework by explicitly modeling duration and pitch alongside discrete unit sequences using a multi-stream Transformer, enabling richer and more natural speech generation. As our corpus expands, this architecture may offer a more expressive

alternative to the basic GSLM pipeline we initially tested. Additionally, the textless speech-to-speech translation framework of Lee et al. (2021) demonstrated that direct speech-to-speech translation is achievable without intermediate text representations, which is particularly relevant for Burushaski given its orthographic instability. Exploring this framework on our data could provide a fully text-free translation pathway. Finally, AudioChatLlama (Fathullah et al., 2023) offers a promising direction for integrating spoken Burushaski input directly with a large language model decoder, enabling zero-shot speech question answering and translation through audio-conditioned LLM inference. As multilingual LLMs continue to scale, this approach may offer meaningful gains even for isolate languages that remain absent from pretraining data, particularly given Burushaski's natural code-switching with Urdu and English.

Evaluation methodology will also be refined. Current assessment relies on standard metrics such as BLEU; future work will complement these with human evaluations by native Burushaski speakers and linguists and will explore character error rate (CER) metrics on any phonemic transcriptions produced as an intermediate step. Targeted evaluation of culturally specific expressions, including idioms and proverbs, will be conducted through structured human evaluation protocols involving native speaker judges. Taken together, these directions position the present corpus and baseline system as a scalable starting point for an ongoing Burushaski speech technology program.

7. Conclusion

This paper presented the first structured effort toward building a curated Burushaski–English parallel speech corpus for direct speech translation. Rather than positioning model performance as the primary contribution, we focused on documenting a reproducible data collection methodology tailored to a severely under-resourced, predominantly oral language. The resulting dataset, though modest in size, establishes an initial benchmark and experimental foundation for future work in Burushaski speech processing.

Our experiments whisper model demonstrates that large pretrained multilingual models can produce semantically coherent English output even when trained on approximately ten hours of parallel data. However, high error rates and frequent lexical substitutions indicate that current model performance remains unstable and far from deployment-ready. In particular, the discrepancy between surface-level metrics (e.g., WER, BLEU) and semantic similarity measures (e.g., BERTScore) suggests that while models capture partial meaning, they struggle with lexical fidelity and culturally specific expressions.

These findings highlight a central challenge in extremely low-resource speech translation: model architecture is not the primary bottleneck—data scale, consistency, and linguistic coverage are. Fine-tuning powerful multilingual models provides measurable gains, but without substantially larger and more diverse parallel corpora, performance will remain constrained.

The broader contribution of this work lies in establishing a practical framework for corpus creation in under-documented languages. Our structured elicitation approach, speaker management protocol, and alignment pipeline provide a template that can be replicated for other low-resource or isolate languages.

Ultimately, sustainable progress in Burushaski speech translation will depend less on incremental architectural modifications and more on systematic, community-centered data development. This work serves as a foundational step in that direction.

8. Bibliographical References

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. (2021). XLS-R: Self-supervised cross-lingual speech representation learning at scale. *arXiv:2111.09296*.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*.

Fathullah, Y., Wu, C., Lakomkin, E., Jia, J., Shangguan, Y., Li, K., Guo, J., Xiong, W., Mahadeokar, J., Kalinli, O., Fuegen, C., and Seltzer, M. L. (2023). AudioChatLlama: Towards general-purpose speech abilities for LLMs. *arXiv:2311.06753.*

Finlayson, N., Marsden, E., and Hawkes, R. (2024). Creating and evaluating corpus-informed word lists for adolescent, beginner-to-low-intermediate learners of French, German, and Spanish. *Language Teaching Research*. December 2024.

Godard, P., Adda, G., Adda-Decker, M., Benjumea, J., Besacier, L., Cooper-Leavitt, J., Kouarata, G.-N., Lamel, L., Maynard, H., Mueller, M., Rialland, A., Schwartz, S., Yvon, F., and Zanon-Boito, M. (2018). A very low resource language speech corpus for computational language documentation experiments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K.,

Salakhutdinov, R., and Mohamed, A. (2021). HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv:2106.07447*.

Kharitonov, E., Lee, A., Polyak, A., Adi, Y., Copet, J., Lakhotia, K., Nguyen, T.-A., Rivière, M., Mohamed, A., Dupoux, E., and Hsu, W.-N. (2021). Text-free prosody-aware generative spoken language modeling. *arXiv:2109.03264*.

Kilgarriff, A., Charalabopoulou, F., Gavrilidou, M., Johannessen, J. H., Khalil, S., Kokkinakis, S. J., Lew, R., Sharoff, S., Vadlapudi, R., and Volodina, E. (2014). Corpus-based vocabulary lists for language learners for nine languages. *Language Resources and Evaluation*, 48(1):123–140.

Lahiri, A. (2018). *Linguistic Fieldwork: A Student Guide*. Cambridge University Press.

Lakhotia, K., Kharitonov, E., Hsu, W.-N., Adi, Y., Polyak, A., Bolte, B., Nguyen, T.-A., Copet, J., Baevski, A., Mohamed, A., and Dupoux, E. (2021). On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

Lee, A., Chen, P.-J., Wang, C., Gu, J., Ma, X., Polyak, A., Adi, Y., He, Q., Tang, Y., Pino, J., and Hsu, W.-N. (2021). Textless speech-to-speech translation on real data. *arXiv:2112.08352*.

Malviya, S., Mishra, R., and Tiwary, U. S. (2016). Structural analysis of Hindi phonetics and a method for extraction of phonetically rich sentences from a very large Hindi text corpus. In *Proceedings of the 2016 Conference of the Oriental Chapter of the International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, Bali, Indonesia. IEEE. https://doi.org/10.1109/ICSDA.2016.7919009

Max Planck Institute for Evolutionary Anthropology. (n.d.). *Field Manuals and Stimulus Materials*. Available at: https://www.eva.mpg.de/lingua/resources/fieldmanuals.php

Lost the Negation or Lost in Negation

Vennela Bairi, Parameswari Krishnamurthy
Language Technology Research Centre, IIIT Hyderabad
vennela.bairi@research.iiit.ac.in, param.krishna@iiit.ac.in

Abstract

Despite negation being one of the core element in any language, it remains a challenging phenomenon for modern Large Language Models(LLMs). Recently, there have been growing efforts to evaluate how models handle negation. However, the existing probing datasets are mostly English-centric. To facilitate evaluation for Indian Languages especially Telugu, which has complex morphological features, we present **NEGTEG** benchmark. This benchmark is a test suite that contains 5 tasks: Negation Detection, Negation Translation, Paraphrase Detection, Sentiment Analysis and Polarity Flipping. The test suite is designed based on strong linguistic analysis and includes annotations of different negation types. This helps us evaluate how models perform across various forms of negation. We use the benchmark to probe the negation handling capabilities of multilingual language models at different levels and our evaluation reveals that most of the models struggle significantly with Telugu negation across all tasks.

Keywords: Negation, Benchmark, Evaluation

1. Introduction

Negation is a fundamental component of natural language and is used to express non-existence, denial, or contradiction. It can change the truth value and can completely flip the meaning of a sentence. Despite its significance, there is ample evidence showing that state-of-the-art NLP models struggle with processing negation effectively. In recent years, there has been a growing effort to evaluate the LLMs' capabilities to handle negation. According to Truong et al. (2022), Khandelwal and Sawant (2020), models are insensitive to the presence of negation. Hossain et al. (2020), Tang et al. (2021), Dobrovolska et al. (2024), Alshargabi et al. (2022), Abuhammad and Ahmed (2024), Gupta and Joshi (2021), reports that the presence of negation significantly impacts downstream task quality like Translation and Sentiment Analysis. Truong et al. (2023), Bhattarai and Erk (2024), She et al. (2023) evaluate how models struggle when asked a logical composition including negation for NLI task. Though there are many existing Benchmarks to evaluate, most of them are inclined towards English. Our work primarily focuses on the systematic evaluation of LLMs' performance for morphologically rich language, Telugu, where negation is not only expressed through words like *no, not, never* as in English, but is also morphologically encoded. This linguistic complexity motivated us to manually curate NEGTEG Benchmark: a test suite of 5 tasks that helps evaluate models' capabilities at different levels of linguistic and logical complexity.

1. **Negation Detection** checks whether models can identify negation cues, evaluating basic syntactic awareness of negation.

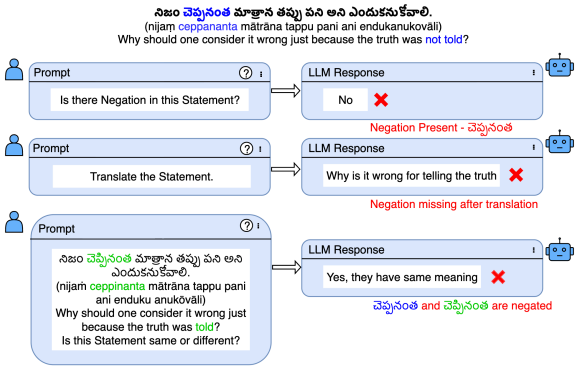


Figure 1: Failures of LLMs' in detecting, preserving, and misinterpretation of semantically opposite forms over negation in Telugu sentences.

2. **Translation** evaluates whether negation is accurately preserved when translating from the source to the target language.
3. **Paraphrase Detection** evaluates whether models can distinguish between semantic equivalence and meaning reversal caused by negation within sentence pairs.
4. **Sentiment Classification** evaluates whether models can correctly determine the sentiment of a sentence in the presence of negation, probing the model's understanding of how negation shifts the sentiment of a sentence.
5. **Polarity Flipping** requires models to rewrite a sentence by flipping the polarity of a specified word or phrase, evaluating whether they can manipulate the negation and not just understand it.

In this paper, we address the lack of resources and evaluation frameworks for negation in Telugu.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 18–31
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

Task	Input / Context	Expected vs Model's Response
Negation Detection	మీ ఆనందంలో పాలు పంచుకొనేందుకు నేను ఉండనే అని దిగులుగా ఉంది. (mī ānaṃdaṃlō pālu pañcukōnēṃduku nēnu uṃḍanē ani digulugā uṃdi) I am sad that I will not be there to share in your happiness.	**Actual:** NEGATED **Model:** NOT_NEGATED
Translation	నిజం చెప్పనంత మాత్రాన తప్పు పని అని ఎందుకనుకోవాలి. (nijaṃ ceppananta mātrāna tappu pani ani endukanukovāli) Why should one consider it wrong just because the truth was not told?	**Expected Output:** Why should one consider it wrong just because the truth was not told? **Model:** Why should one consider it a wrong act just for telling the truth?
Paraphrase Detection	**S1:** నాకు నచ్చని పని చేసినందుకు నా అహం దెబ్బ తింది. (nāku naccani pani cēsinanduku nā ahaṃ debba tiṃdi.) My ego was hurt because I did work that I did not like. **S2 (Paraphrase):** నాకు ఇష్టంలేని పని చేయడంవల్ల నా అహంకారానికి భంగం కలిగింది. (nāku iṣṭaṃlēni pani cēyaḍaṃvalla nā ahaṃkārāniki bhaṃgaṃ kaligindi.) My pride was wounded by doing work I disliked. **S2 (Negated):** నాకు నచ్చని పని చేసినందుకు నా అహం దెబ్బ తినలేదు. (nāku naccani pani cēsinanduku nā ahaṃ debba tinalēdu.) My ego was not hurt by doing work I disliked.	*For S2 Paraphrase:* **Actual:** PARAPHRASE **Model:** PARAPHRASE *For S2 Negated:* **Actual:** NEGATED **Model:** PARAPHRASE
Sentiment Classification	స్మోక్ ఫ్రీ నినాదానికి ఫైవ్ స్టార్ హోటళ్లకు కూడా మినహాయింపు లేదు. (smōk free ninādāniki fayiv sṭār hōṭaḷlaku kūḍā mināhāyiṃpu lēdu) Even five-star hotels are not exempt from the smoke-free initiative.	**Actual:** POSITIVE **Model:** NEGATIVE
Polarity Flipping	నేను ఆమెని చూడనట్టు నటిస్తూ తప్పించుకున్నాను. (nēnu āmēni cūḍanattu naṭistū tappiñcukunnānu) I pretended not to see her and escaped. **Verb to flip:** తప్పించుకున్నాను (tappiñcukunnānu) escaped	**Expected Output:** నేను ఆమెని చూడనట్టు నటిస్తూ తప్పించుకోలేదు. (tappiñcukōlēdu) didn't escape **Model:** నేను ఆమెని చూడనట్టు నటిస్తూ తప్పించుకున్నాను. (tappiñcukunnānu) escaped

Table 1: Example instances from each task in the benchmark, with Expected and Model's Response

We release the benchmark publicly[1]. Table 1 illustrates a sample for each task. To best of our knowledge, this is the first work on evaluating LLMs on downstream tasks for negation in Telugu. To this end, our major contributions are:

1. We introduce a manually extracted and verified comprehensive list of Telugu negative morphemes, covering the full range of negation patterns possible in Telugu.
2. We present **NEGSEED**, a manually curated set of example sentences for each of the above morphemes, serving as a foundational resource for Telugu negation.
3. We present **NEGTEG**, a test suite of 5 tasks designed to systematically probe LLMs' capabilities to handle negation across varying levels of linguistic and logical complexity.
4. We evaluate a diverse set of LLMs on this benchmark, providing a thorough analysis of how effectively the models process negation in Telugu.

[1] https://github.com/VennelaBairi/NEGTEG

2. Related Work

Several works have investigated the negation handling capabilities of models by creating task-specific datasets and analyzing model outputs.

Negation Detection: Studies have exposed how state-of-the-art NLP models fail to find negation cues. Khandelwal and Sawant (2020) reviewed previous literature including rule-based systems, ML classifiers, CRF models, CNNs, BiLSTMs. Okpala et al. (2022) demonstrated methods for detecting negations in a sentence by evalu-

ating the lexical structure of the text via word-sense disambiguation. Martinis et al. (2024) reviewed how negation detection plays an important role in understanding clinical documentation. Other works highlighted the importance of detecting the scope of negation for downstream tasks such as sentiment analysis (Makkar et al., 2024) and also in Information Extraction (Weller et al., 2024).Shah and Pareek (2024) conducted a survey on negation detection in Hindi, exploring both BERT-based and hybrid rule-based approaches, finding that the complex syntactic structure of Hindi differs greatly from English, making the direct application of English models inadequate.

Translation: Fancellu and Webber (2015) conducted an error analysis while translating Chinese into English, which involves negation. They found that identifying the negation cue right is easier than translating the event and scope correctly. Hossain et al. (2020) reported that negation is often mistranslated and has the potential to cause loss of information. Tang et al. (2021) similarly reports that that negations are under-translated and according to a qualitative analysis by Alshargabi et al. (2022), students faced difficulty translating sentences containing double negation

Paraphrase Detection: Work by Ettinger (2020) on Pretrained Language models like BERT and ELMo and extended work by Truong et al. (2023) on autoregressive models state that models are insensitive to the contextual impacts of negation. Anschütz et al. (2023) designed the CANNOT Dataset for paraphrased or negated classification and fine-tuned a sentence transformer to improve their negation sensitivity. Vahtola et al. (2022) finds that LLMs underestimate the impact of negations on how much they change the meaning of a sentence. Rezaei and Blanco (2024) finds that paraphrasing in affirmative terms improves the performance of negation understanding. The benchmarks constructed by Hartmann et al. (2021) and So et al. (2025) evaluate how the models understand and reason in the NLI task.

Sentiment analysis: Despite of having positive or negative keywords, the sentiment can be easily flipped by inserting negations, making sentiment analysis challenging. The study by Mamatha Mylarappa et al. (2023), Ilmawan et al. (2024), Mukherjee et al. (2021) highlight the importance of considering negation in the text to improve the performance of sentiment analysis. Cruz et al. (2016)'s work on negation and speculation information in review texts says that the correct identification of cues and scopes is vital for the task of sentiment analysis and opinion mining systems.

Polarity Flipping: Experiments conducted by Chen et al. (2023) and Ghasiya and Sasahara (2026) reveal that LLMs frequently fail to generate valid sentences grounded in negative common sense knowledge. They further find that though guided prompts improve the quality of generated negatives, they often generate ambiguous statements, or statements with negative keywords but a positive meaning.

Negation	Example	Example Sentence
Explicit	కాదు (kādu) is not	అతను విద్యార్థి కాదు. (atanu vidyārthi kādu.) He is not a student.
Morphological	వెళ్ళకు (veḷḷaku) do not go	అక్కడికి వెళ్ళకు. (akkadiki veḷḷaku.) Do not go there.
Lexical	అసాధ్యం (asādhyam) impossible	ఇది అసాధ్యం. (idi asādhyam.) This is impossible.

Table 2: Types of negation in Telugu with illustrative examples across four categories.

Pretrained Language Models like BERT, have been shown to fail in distinguishing between negated and non-negated sentences (Hosseini et al., 2021), and automatic evaluation metrics such as BERTScore are equally insensitive to negation (Hanna and Bojar, 2021).

These observations, along with the lack of resources for Indian languages, collectively motivate the creation of a benchmark for evaluating downstream tasks in Telugu.

3. Benchmark Creation

3.1. Base Corpora Collection

For creating the benchmark we used Telugu Books dataset available on Kaggle[2] which contains 25,000 books. This extensive collection of original Telugu books in various genres ensures that the benchmark captures the authentic nuances and natural flow of the language. The Indic Sentence Tokenizer[3] was applied on the books dataset to extract sentences.

3.2. Negation Types and Coverage

Languages express negation in different forms and types. When separate negative words are used, it is called Explicit Negation. When specific morphemes are added to the words, it is called Morphological Negation. Other ways include Lexical Negation, where certain words carry negative meaning, such as impossible or unlikely. Table 2 illustrates types of negation with examples.

Telugu, being a morphologically rich language, expresses negation not only through Explicit, Lex-

[2] https://www.kaggle.com/datasets/sudalairajkumar/telugu-nlp

[3] https://indic-nlp-library.readthedocs.io/en/latest/indicnlp.tokenize.html

ical forms, but predominantly by attaching negative morphemes directly to words, mostly verbs. For example, for the verb చెప్పు (cheppu -to say), adding negative morphemes such as _అకు (_aku) and _అను (_anu) results in forms like చెప్పకు (ceppaku) (do not say) and చెప్పను (cheppanu) ("I will not say").

Morpheme	Example
అని_అట్లు (ani_aṭlu)	చెప్పనట్లు (ceppanaṭlu) as if not said'
అని (ani)	చెప్పని (ceppani) that which is not said
అ_లేదు (a_lēdu)	చెప్పలేదు (ceppalēdu) did not say
అడం_లేదు (aḍaṃ_lēdu)	చెప్పడంలేదు (ceppaḍaṃlēdu) is not saying
అని_అప్పుడు (ani_appuḍu)	చెప్పనప్పుడు (ceppanappuḍu) when not saying
అని_అంత (ani_anta)	చెప్పనంత (ceppananta) to the extent of not saying
అ_కూడదు (a_kūḍadu)	చెప్పకూడదు (ceppakūḍadu) must not say

Table 3: Telugu Negation Morphemes illustrated with the root verb చెప్పు (cheppu) 'to say'.

In Telugu there are approximately 800 distinct morpheme combinations (Dasari et al., 2023). We examined this complete set to identify morphemes that carry a negative meaning, as shown in Table 3. Through manual extraction and cross-verification, we identified 104 negative morphemes. Of these, we used 94 in this study, excluding 10 traditional forms that rarely appear in contemporary Telugu.

Building a dataset with this extensive negative morpheme coverage required a hybrid collection strategy. As a first step, we applied a Morphological Analyzer Dasari et al. (2023) to the tokenized sentences. This tool decomposes each word form into its constituent morphemes, providing seven fields of information: Root, Lexical category (lcat), Gender (gen), Number (num), and Person (per), Case, Suffix(when lcat is noun)/Tense-Aspect-Mood(TAM) (when lcat is verb). Table 4 illustrates this analysis for two example words.

Word	Root	lcat	Gen	Num	Per	Case	TAM
చెప్పలేదు	చెప్పు	V	any	any	any	–	అ_లేదు
చెప్పవు	చెప్పు	V	n	pl	3	–	అ

Table 4: Morphological analysis of example negative verb forms (V- Verb and pl- plural)

Using the morphological decomposition produced by this analyzer, we matched the TAM field of each tokenized word against our list of 94 negative morphemes. This automatic extraction identified approximately 3,000 sentences that covered 54 of the 94 negative morphemes. From these, we selected 270 sentences covering 5 sentences per morpheme, ensuring diversity in sentence structure and coverage across Negative Polarity Items (NPIs) DILIP and KUMAR (2019) (Table 10). All selected sentences were reviewed by native Telugu speakers to verify the grammatical correctness and naturalness.

The remaining 40 morphemes were underrepresented in the extracted data, either because they are relatively uncommon or because their morphological variants made them difficult to capture through pattern matching. For these, we employed human annotators to generate sentences. Annotators were provided with 800 sentences from the Telugu Books dataset as reference and were asked to modify them to incorporate the missing negative morphemes. They were also given detailed specifications for each morpheme, including its morphological structure and example words. This manual process yielded 200 sentences covering all 40 remaining morphemes.

Through this combined process of automatic extraction (270 sentences), human generation (200 sentences), followed by validation by native speakers, we compiled a final set of 470 negated sentences, which we refer to as the **NEGSEED** set. This set serves as the foundation for the datasets used in each test suite of our benchmark. NEGSEED is designed to comprehensively evaluate whether LLMs can handle the full morphological and syntactic range of Telugu negation. Since sentences are sourced from authentic Telugu literature, NEGSEED reflects natural linguistic diversity rather than artificially constructed examples. It serves as a reusable reference that can be extended to other domains beyond book-based text, including conversational, web, and task-specific corpora.

3.3. Test Suite for Benchmark

To evaluate LLMs' capabilities in understanding and handling negation, we designed 5 tasks at increasing levels of complexity, starting from whether models can detect the presence of negation, how negation affects downstream tasks, whether they could differentiate between the negated and non-negated pairs, how well they reason and understand when negation is involved, how well models can flip the polarity of a sentence. The following subsections describe how the data for each task was prepared and Table 1 illustrates a sample of each task.

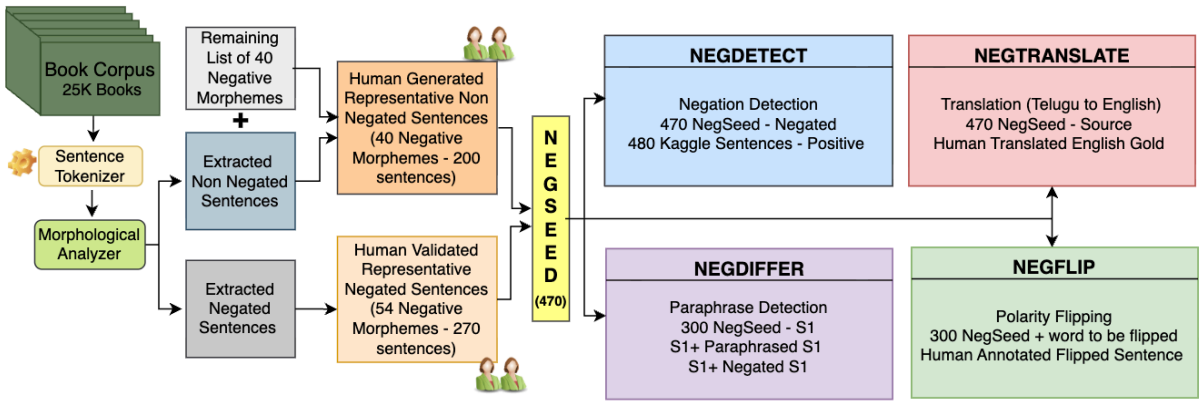


Figure 2: Pipeline for building NEGTEG, a negation-focused evaluation benchmark. NegSeed is first constructed through corpus filtering and human validation, and subsequently expanded into four test suites: NEGDETECT, NEGTRANSLATE, NEGDIFFER, and NEGFLIP.

3.3.1. NEGDETECT: Negation Detection

Our primary motive to include this task in the test suite is to check whether the LLMs can at least detect the presence of negation. This capability is a prerequisite for more complex tasks. The test set has 950 sentences in total with the labels NEGATED and NON NEGATED. It consists of 470 negated sentences from the NEGSEED set and 480 non-negated sentences drawn from the same Telugu Books corpus to ensure class balance. All non-negated sentences were validated by annotators to confirm the absence of negative morphemes and explicit negation words. The two classes are defined as follows:

- **NEGATED:** Sentences that contain explicit negative words and Negative morphemes.
- **NON-NEGATED:** Sentences that have Lexical negatives and other non-negated sentences.

3.3.2. Translation

Translation is a very common task, but this test set evaluates how well LLMs and translation models preserve negation during translation. We used the NEGSEED set as input. Each sentence was translated into English by human translators. They were given explicit instructions to preserve the negation in the target language rather than omitting or paraphrasing it, ensuring that the negation is explicitly reflected in the English translation and not merely implied by the overall meaning.

3.3.3. Paraphrase Detection

This task evaluates whether the model can understand how negation alters the meaning of a sentence and the logical relationship between two sentences. For this test set, a subset of 300 sentences from NEGSEED served as Sentence 1. For each Sentence 1, two versions of Sentence 2 were manually generated as follows:

- **NEGATED S2:** A sentence whose meaning is reversed by flipping the polarity of the whole sentence or a specific part of Sentence 1.
- **PARAPHRASED S2:** A sentence that retains the same meaning as Sentence 1 but uses different words or structures.

Given Sentence 1 and Sentence 2, the model should predict the correct label NEGATED or PARAPHRASED. This task requires the model to understand how negation alters the meaning and logical relationship between statements.

3.3.4. Sentiment Task

To construct the benchmark, we utilized two publicly available Telugu sentiment datasets: ACTSA and SentiRama[4]. We utilized the complete ACTSA dataset (Mukku and Mamidi, 2017). From SentiRama, we consider only the Books Reviews and Product Reviews subsets.

To keep the analysis straightforward, we considered only two classes: POSITIVE and NEGATIVE. Samples labeled as NEUTRAL were discarded, as binary classification provides a clearer understanding for evaluating how negation shifts sentiment polarity. This task directly tests whether models understand the effects of negation on sentiment. A model may correctly predict positive sentiment for a word like *'good'*, but must also recognize that the addition of negation, as in *'not good'*, reverses the overall sentiment. A model that fails to account for negation will likely misclassify such sentences.

[4]SentiRama:https://ltrc.iiit.ac.in/showfile.php?filename=downloads/sentiraama/

This serves as a strong indicator of a model's understanding of negation in a downstream task.

3.3.5. Polarity Flipping

This task evaluates the model's ability to generate semantically opposite sentences through controlled polarity flipping. Since a sentence can produce multiple valid negations depending on which constituent is negated, we explicitly specify the word or phrase whose polarity is to be flipped (Table 5). Each sample in the dataset consists of an input sentence, the target word or phrase to be flipped, and a human-annotated reference sentence with reversed polarity. Two annotators independently annotated 150 samples each, followed by cross-validation to ensure consistency in target selection and reference quality.

Sentence: నాకు నచ్చని పని చేసినందుకు నా అహం దెబ్బ తింది. (nāku naccani pani cēsinaṃduku nā ahaṃ debba tindi) My ego was hurt because I did work that I did not like.	
Word to be Flipped	**Polarity Flipped Sentence**
చేసినందుకు (cēsinaṃduku) *'because I did'*	నాకు నచ్చని పని చేయనందుకు నా అహం దెబ్బ తింది. (nāku naccani pani cēyanaṃduku nā ahaṃ debba tindi) My ego was hurt because I did not do the work I disliked.
నచ్చని (naccani) *'did not like'*	నాకు నచ్చిన పని చేసినందుకు నా అహం దెబ్బ తింది. (nāku naccina pani cēsinaṃduku nā ahaṃ debba tindi) My ego was hurt because I did work that I liked.

Table 5: Multiple valid negations of the original sentence, each targeting a different constituent.

4. Evaluation and Results

4.1. Evaluating Negation Detection

This experiment evaluates whether the models can identify the presence of explicit negation and negative morphemes in a sentence. This is a preliminary, foundational check to assess whether the models are sensitive to negation cues or tend to overlook them. Models are explicitly prompted to classify a sentence as NEGATIVE only if it contains explicit negative words or negative morphemes. Sentences conveying negation through lexical as well as all affirmative sentences, must be classified as NON_NEGATED(Appendix B.1).

Table 6 reports Accuracy, Precision and Recall for both NEGATED and NON_NEGATED classes, and Macro F1. Larger models such as GPT-OSS 120b, GPT-OSS 20b, and Llama-3.3-70b-versatile tend to achieve higher NonNegRecall but lower NegRecall. This indicates that out of all negatives, fewer negated sentences are correctly identified compared to non-negated ones. However, smaller models with competitive accuracy, such as Meta-Llama-3-8B, Nemotron-Cascade-8B, and Qwen3-8B, exhibit higher NegRecall than NonNegRecall, indicating a tendency to predict negation more liberally. Models with relatively lower accuracy show even more heavily skewed predictions, collapsing almost entirely towards one class. Overall, most models struggle to correctly identify both negated and non-negated sentences, highlighting the difficulty of recognizing negation in Telugu.

4.2. Evaluating Translation

We gave the models a simple prompt: "*Translate the following Telugu sentence to English.*" This prompt, without any additional guidance regarding negation, is used to evaluate how negation is handled in a downstream task without giving any explicit instructions about the negation. The idea behind this test set is to examine whether the negation is semantically understood in a downstream task and translated from source to target language.

Along with the LLMs used for all other tasks, we also evaluated dedicated translation models including Google Translate, Rotary-Indictrans2, Bhashaverse (Mujadia and Sharma, 2024) and Sarvam Translate. Although standard translation metrics, such as BLEU, chrF++, METEOR, and COMET (Table 7), measure the overall quality of the translation, they do not specifically consider whether the negation present in the source sentence is preserved after translation. Therefore, we employed human evaluators to validate if negation is preserved after translating. The guidelines given to evaluators are in Appendix A.1.

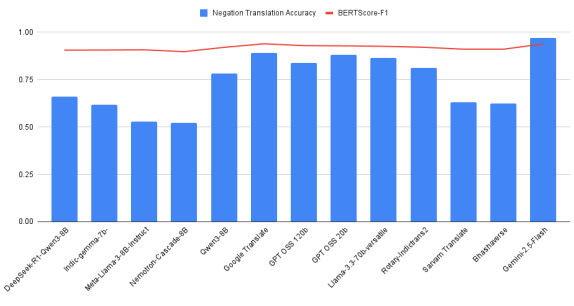


Figure 3: Human-Evaluated Negation Preservation Percentage Across Translation Models

Model	Accuracy	NegPrecision	NonNegPrecision	NegRecall	NonNegRecall	MacroF1
DeepSeek-R1-Qwen3-8B	57.53	53.93	83.49	95.94	20.04	50.69
Indic-gemma-7b-Navarasa	50.63	66.67	50.58	0.43	99.79	33.99
Meta-Llama-3-8B-Instruct	83.47	83.09	83.86	83.62	83.33	83.47
Nemotron-Cascade-8B	81.58	79.32	84.12	84.89	78.33	81.57
Qwen3-8B	79.26	74.86	85.29	87.45	71.25	79.15
Gemma-7b-it	59.16	80.60	55.64	22.98	94.58	52.91
GPT-OSS 120b	94.42	98.60	90.98	90.00	98.75	94.40
GPT-OSS 20b	91.54	97.85	85.09	87.02	97.48	91.49
Llama-3.3-70b-versatile	78.35	97.01	61.94	69.15	95.93	78.01
Sarvam-1	64.74	89.47	59.31	32.55	96.25	60.56
Gemini-2.5-flash	99.37	99.36	99.36	99.38	99.38	99.37

Table 6: Performance of LLMs on NEGDETECT, reporting Accuracy, Precision, Recall, and Macro F1 for NEGATED and NON_NEGATED classes.

Model	**BLEU**	**chrF++**	**METEOR**	**COMET**
DeepSeek-R1-Qwen3-8B	12.48	34.72	0.39	0.68
Indic-gemma-7b-Navarasa	13.10	33.80	0.39	0.69
Meta-Llama-3-8B-Instruct	13.12	35.58	0.40	0.69
Nemotron-Cascade-8B	10.48	31.67	0.36	0.65
Qwen3-8B	19.05	41.07	0.47	0.74
Google Translate	33.47	53.27	0.60	0.80
GPT OSS 120b	24.60	47.19	0.55	0.76
GPT OSS 20b	23.41	46.06	0.54	0.76
Llama-3.3-70b-versatile	22.95	45.26	0.51	0.76
Rotary-Indictrans2	18.43	40.08	0.46	0.74
Sarvam Translate	11.65	33.13	0.38	0.71
Bhashaverse	15.61	35.88	0.41	0.68
Gemini-2.5-Flash	28.57	51.33	0.59	0.80

Table 7: Performance of LLMs and Translation models on the Translation task, evaluated using BLEU, chrF++, METEOR, and COMET metrics.

From Figure 3, it is evident that models are unable to accurately translate negation. Notably, while the difference in BERTScore across models is not substantial, the negation preservation accuracy varies significantly, suggesting that standard metrics alone may not capture how well negation is preserved (Hosseini et al., 2021). Our manual qualitative analysis reveals some common error patterns: the negation is completely omitted in the translation, it is mistranslated, or the negation is dropped from the intended word or phrase and incorrectly applied to another (Appendix Table 11).

4.3. Evaluating Paraphrase Detection

In the Negation detection task, the objective was relatively straightforward: given a single sentence, determine whether it contains negation. In contrast, the paraphrase detection task is more challenging. Here, the model is provided with sentence pairs, either a sentence paired with its negated version or a sentence paired with its paraphrased version. The models are prompted to determine whether a given sentence pair conveys the same meaning or opposite meaning (at least in part) which helps distinguish between semantic equivalence and contradiction introduced by negation.

Although overall accuracy scores are low for several small models, the performance drop is primarily driven by errors in negated pairs. Recall analysis (Table 8) shows that paraphrased instances are identified with relatively high recall, whereas negated pairs are detected much less reliably, indicating difficulty in recognizing meaning reversal caused by negation. Even when a negated sentence appears alongside its original, models often fail to capture the semantic contradiction. However, Larger models such as GPT-OSS 120B and 20B, and Llama-3.3-70B exhibit higher NegRecall than ParaRecall, while smaller models show the opposite trend. This suggests that smaller models tend to treat most sentence pairs as semantically equal though they are not.

4.4. Evaluating Sentiment Classification

We first categorize the sentiment test set into sentences containing negation and sentences without negation, using a morphological analyzer (Dasari et al., 2023) to identify the presence of explicit negative words or negative morphemes. This categorization allows us to directly compare model perfor-

Model	Accuracy	ParaPrecision	NegPrecision	ParaRecall	NegRecall
DeepSeek-R1-Qwen3-8B	68.85	69.58	68.09	69.06	68.63
Indic-gemma-7b-Navarasa	53.89	52.24	61.86	86.88	21.13
Meta-Llama-3-8B-Instruct	51.77	50.82	86.67	99.29	4.58
Nemotron-Cascade-8B	62.19	57.00	93.75	98.23	26.41
Qwen3-8B	83.22	77.10	92.76	94.33	72.18
Gemma-7b-it	49.82	49.82	0.00	100.00	0.00
GPT-OSS 120b	88.16	90.26	86.29	85.46	90.85
GPT-OSS 20b	89.17	91.67	86.96	86.12	92.20
Llama-3.3-70b-versatile	87.10	96.86	80.76	76.60	97.54
Telugu-Llama2-7B-Instruct	49.82	49.82	0.00	100.00	0.00
Gemini-2.5-Flash	97.17	98.19	96.21	96.10	98.24

Table 8: Performance of LLMs on NEGDIFFER, reporting Accuracy, Precision, and Recall for PARAPHRASED and NEGATED classes.

	Flip Quality Metrics				Similarity Metrics		
Model	Correct (%)	(5,4) (%)	3 (%)	(2,1) (%)	BLEU	BERTScore	CharEdit$_{norm}$
DeepSeek-R1-Qwen3-8B	40.15	22.78	21.62	55.60	0.6133	0.9007	0.3044
Indic-gemma-7b-Navarasa	20.45	15.24	10.41	74.35	0.5562	0.8966	0.2969
Meta-Llama-3-8B-Instruct	18.42	10.90	7.89	80.45	0.6280	0.8975	0.2794
Qwen3-8B	58.58	42.54	22.76	34.70	0.6640	0.9216	0.2169
Gemma-7b-it	1.14	0.76	1.14	98.11	0.6209	0.9019	0.2836
GPT-oss-120b	58.21	47.76	20.15	32.09	0.7056	0.9381	0.1641
GPT-oss-20b	57.98	47.47	18.29	34.24	0.6896	0.9325	0.1812
Llama-3.3-70b-versatile	70.00	55.93	23.33	20.74	0.6898	0.9282	0.1860
Gemini-2.5-Flash	86.54	78.08	13.08	8.85	0.7393	0.9492	0.1367

Table 9: Performance of LLMs on NEGFLIP, reporting Flip Quality metrics (percentage of correctly flipped instances and distribution of Meaning scores ((5,4), 3, and (2,1))) and similarity metrics (BLEU, BERTScore, and normalized character edit distance).

mance across the sentences that contain explicit and morphological negations and others. From Figure 4, it is observed that accuracy is higher for sentences without negation. This suggests that negation introduces additional complexity.

From Appendix Table 12, it is evident that, recall for the positive sentiment class under negated conditions is consistently low across most models, indicating that models struggle to correctly identify positive sentiment when negation is present. This is likely because negation words pull the model toward predicting negative sentiment directly, rather than understanding the compositional meaning of the negated sentence. Conversely, recall for the negative sentiment class under negated conditions remains high, suggesting that models associate negation cues directly with negative sentiment rather than reasoning about the actual sentiment of the sentence.

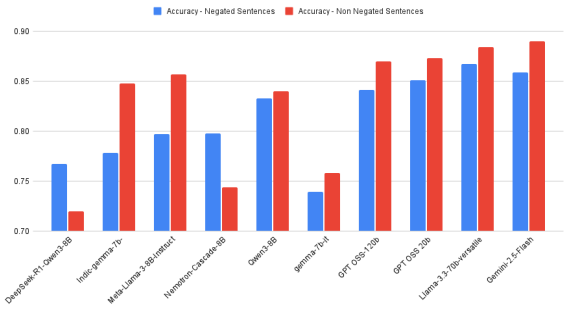


Figure 4: Comparison of overall sentiment classification accuracy on sentences containing negation and sentences without negation.

4.5. Evaluating Polarity Flipping

This experiment assesses whether models can generate a negated variant of a given sentence by flipping the meaning of a specified target word or phrase while preserving the rest of the sentence's structure and semantics. The human evaluation guidelines for this task are detailed in the Appendix A.2. The Results in Table 9 show that Polarity Flipping is the hardest task in this benchmark. Even large models like GPT-OSS 120b and GPT-OSS 20b achieve only 58% correctness, with only Gemini-2.5-Flash, Llama-3.3-70b-versatile performing reasonably well. However, models that fail to correctly flip the polarity still achieve high BERTScores, meaning the generated sentences look similar to the input but do not carry the intended opposite meaning highlighting the inadequacy of automated metrics.

5. Conclusion

In this work, we present **NEGTEG**, a comprehensive benchmark to evaluate negation handling in Telugu on five tasks: Negation Detection, Negation Translation, Paraphrase Detection, Sentiment classification, and Polarity flipping. Our evaluation of models reveals that negation in Telugu remains a persistent challenge even for state-of-the-art models. Models tend to be biased toward non-negative predictions, struggle to preserve negation in translation, and fail to distinguish semantically inverted sentence pairs in paraphrase detection. Furthermore, models struggle with downstream tasks in the presence of negation, particularly in sentiment classification and polarity flipping. A recurring observation is that automated metrics remain high even when models fail semantically, underscoring the need for human evaluation. Overall, larger models consistently outperform smaller ones, yet even the strongest models struggle with deep compositional understanding of negation in Telugu.

Limitations and Future Work

The benchmark covers only Telugu and does not cover other Indian languages, limiting its generalizability. Human evaluation, while necessary for translation and polarity flipping tasks, is inherently subjective and may introduce annotator bias despite the provided guidelines. Finally, the benchmark evaluates models in a zero-shot setting, and performance may vary with few-shot prompting or fine-tuning on Telugu-specific data. Although our evaluation reveals that models are insensitive to negation in Telugu, the lack of sufficient training data prevented us from fine-tuning the models to address this limitation.

As future work, we plan to extend NEGTEG to other Indian languages, especially Dravidian languages such as Tamil, Kannada, and Malayalam. Since these languages are agglutinative and morphologically rich like Telugu, this will help us evaluate whether the trends observed in Telugu hold across similar languages.

Acknowledgements

We sincerely thank Deepika Ketha (M.Sc. Computer Science) and T V N Prasanthi (B.Tech), both native Telugu speakers, for their valuable contributions to this work. They have extensive experience in creating linguistic benchmark datasets and working as translators, which was invaluable to the annotation and evaluation process. Their efforts in building the benchmark dataset, cross-verifying annotations, and evaluating model outputs were essential to ensuring the quality and reliability of NEGTEG. We also thank Nagaraju Vuppala for his support with the morphological analysis tool.

Ethical Considerations

We release NEGTEG publicly to support further research on negation in Telugu. The benchmark was built using a books dataset sourced from Kaggle, and we do not intend to republish the original data. It is used solely for research purposes.

References

Ahmed Suliman Abuhammad and Mahmoud Ali Ahmed. 2024. Negation detection techniques in sentiment analysis: A survey. *Iraqi Journal of Science*, pages 1060–1069.

Sahar Abdulsalam Alshargabi, Dina Fahmi Kamil, and Ali Hussein Hazem. 2022. A linguistic study of english double negation and its realization in arabic. *Studies in English Language and Education*, 9(3):1148–1169.

Miriam Anschütz, Diego Miguel Lozano, and Georg Groh. 2023. This is not correct! negation-aware evaluation of language generation systems. *arXiv preprint arXiv:2307.13989*.

Gunjan Bhattarai and Katrin Erk. 2024. To learn or not to learn: Replaced token detection for learning the meaning of negation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16237–16250.

Jiangjie Chen, Wei Shi, Ziquan Fu, Sijie Cheng, Lei Li, and Yanghua Xiao. 2023. Say what you mean! large language models speak too positively about negative commonsense knowledge. *arXiv preprint arXiv:2305.05976*.

Noa P Cruz, Maite Taboada, and Ruslan Mitkov. 2016. A machine-learning approach to negation and speculation detection for sentiment analysis. *Journal of the Association for Information Science and Technology*, 67(9):2118–2136.

Priyanka Dasari, Abhijith Chelpuri, Nagaraju Vuppala, Mounika Marreddy, Parameshwari Krishnamurthy, and Radhika Mamidi. 2023. Transformer-based context aware morphological analyzer for telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 25–32.

Mayuri J DILIP and Rajesh KUMAR. 2019. Negative polarity items in telugu. *Acta Linguistica Asiatica*, 9(1):9–28.

Svitlana Romanivna Dobrovolska, Mariana Bohdanivna Opyr, and Svitlana Bohdanivna Panchyshyn. 2024. The category of negation in scientific and technical discourse (translation aspect).

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Federico Fancellu and Bonnie Webber. 2015. Translating negation: A manual error analysis. In *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, pages 2–11. Association for Computational Linguistics.

Piyush Ghasiya and Kazutoshi Sasahara. 2026. From generation to detection: Leveraging empirically derived linguistic hints for llm-based fake news detection.

Itisha Gupta and Nisheeth Joshi. 2021. A review on negation role in twitter sentiment analysis. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, 16(4):1–19.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.

Mareike Hartmann, Miryam de Lhoneux, Daniel Hershcovich, Yova Kementchedjhieva, Lukas Nielsen, Chen Qiu, and Anders Søgaard. 2021. A multilingual benchmark for probing negation-awareness with minimal pairs. In *CoNLL 2021-25th Conference on Computational Natural Language Learning*, pages 244–257. Association for Computational Linguistics.

Md Mosharaf Hossain, Antonios Anastasopoulos, Eduardo Blanco, and Alexis Palmer. 2020. It's not a non-issue: negation as a source of error in machine translation. *arXiv preprint arXiv:2010.05432*.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1301–1312.

Lutfi Budi Ilmawan, Didik Dwi Prasetya, et al. 2024. Negation handling for sentiment analysis task: approaches and performance analysis. *International Journal of Electrical & Computer Engineering (2088-8708)*, 14(3).

Aditya Khandelwal and Suraj Sawant. 2020. Negbert: a transfer learning approach for negation detection and scope resolution. In *Proceedings of the twelfth language resources and evaluation conference*, pages 5739–5748.

Kartika Makkar, Pardeep Kumar, Monika Poriye, and Shalini Aggarwal. 2024. Improving sentiment analysis using negation scope detection and negation handling. *International Journal of Computing and Digital Systems*, 16(1):239–247.

Shiva Kumar BN Mamatha Mylarappa, Thriveni J Gowda, and Venugopal K Rajuk. 2023. Enhanced negation handling for sentiment analysis on twitter using deep neural networks. *Indonesian Journal of Electrical Engineering and Computer Science*, 32(3):1736–1745.

Maria Chiara Martinis, Chiara Zucco, and Mario Cannataro. 2024. Negation detection in medical texts. In *International Conference on Computational Science*, pages 75–87. Springer.

Vandan Mujadia and Dipti Misra Sharma. 2024. Bhashaverse: translation ecosystem for indian subcontinent languages. *arXiv preprint arXiv:2412.04351*.

Partha Mukherjee, Youakim Badr, Shreyesh Doppalapudi, Satish M Srinivasan, Raghvinder S Sangwan, and Rahul Sharma. 2021. Effect of negation in sentences on sentiment analysis and polarity detection. *Procedia Computer Science*, 185:370–379.

Sandeep Sricharan Mukku and Radhika Mamidi. 2017. Actsa: Annotated corpus for telugu sentiment analysis. In *Proceedings of the first workshop on building linguistically generalizable NLP systems*, pages 54–58.

Izunna Okpala, Guillermo Romera Rodriguez, Andrea Tapia, Shane Halse, and Jess Kropczynski. 2022. A semantic approach to negation detection and word disambiguation with natural language processing. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*, pages 36–43.

MohammadHossein Rezaei and Eduardo Blanco. 2024. Paraphrasing in affirmative terms improves negation understanding. *arXiv preprint arXiv:2406.07492*.

Nirja Shah and Jyoti Pareek. 2024. Optimized hindi negation detection using a hybrid rule-based and bert model. In *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, pages 544–550. IEEE.

Jingyuan S She, Christopher Potts, Samuel Bowman, and Atticus Geiger. 2023. Scone: Benchmarking negation reasoning in language models with fine-tuning and in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1803–1821.

Yeonkyoung So, Gyuseong Lee, Sungmok Jung, Joonhak Lee, JiA Kang, Sangho Kim, and Jaejin Lee. 2025. Thunder-nubench: A benchmark for llms' sentence-level negation understanding. *arXiv preprint arXiv:2506.14397*.

Gongbo Tang, Philipp Rönchen, Rico Sennrich, and Joakim Nivre. 2021. Revisiting negation in neural machine translation. *Transactions of the Association for Computational Linguistics*, 9:740–755.

Thinh Truong, Timothy Baldwin, Trevor Cohn, and Karin Verspoor. 2022. Improving negation detection with negation-focused pre-training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4188–4193.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. Language models are not naysayers: an analysis of language models on negation benchmarks. *arXiv preprint arXiv:2306.08189*.

Teemu Vahtola, Mathias Creutz, and Jörg Tiedemann. 2022. It is not easy to detect paraphrases: Analysing semantic similarity with antonyms and negation using the new semantoneg benchmark. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 249–262.

Orion Weller, Dawn Lawrie, and Benjamin Van Durme. 2024. Nevir: Negation in neural information retrieval. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2274–2287.

Appendix

A. Human Evaluation Guidelines

This appendix documents the guidelines provided to human annotators for evaluation tasks. To ensure consistency and reliability, annotators were briefed on these guidelines before beginning the annotation process.

A.1. Guidelines for Negation Preservation in Translation

Human Evaluation Guidelines for Negation Preservation after Translation
Preserved:
- The negation present in the source sentence is correctly translated.
- The scope of the negated verb or phrase is accurately preserved.

Not Preserved:
- The negation is completely omitted in the translation.
- The negation incorrectly modifies a different word or scope.
- The negation is mistranslated due to incorrect morphological agreement in gender or tense.

A.2. Guidelines for Polarity Flipping

Human Evaluation Guidelines for Polarity Flipping
Polarity Check:
- Is the polarity of the specified target correctly flipped? (Yes / No)

Meaning Preservation Rating:
- 5 – Perfect polarity reversal; meaning matches the reference.
- 4 – Meaning is mostly preserved with minor variation.
- 3 – Minor meaning change compared to the reference.
- 2 – Major change in meaning; polarity not correctly flipped.
- 1 – Completely irrelevant or incorrect generation.

B. Prompts Used for Model Inference

B.1. Negation Detection Prompt

You are a Telugu linguistics expert.
Your task is to analyze the following Telugu sentences and determine if they contain negation.
In Telugu, negation can be explicit (using words like కాదు, లేదు, వద్దు) or morphological negation where morphemes attached to verbs or clauses

(using suffixes like _లేదు, _అకపోతే, _అక, _అకు, _అన్_ఏ_లేదు, _అ_లెండి, and many more) or lexical (అసాధ్యం, అసంభవం, అన్యాయం, నిరాశ)
Rules:
- NEGATED (if explicit negative or negated morpheme is identified)
- NOT_NEGATED (if no explicit grammatical negation is present, regardless of whether the sentence contains lexical negativity.)
Sentence: "{sentence}"
Return ONLY JSON:
{"label":"NEGATED"} or {"label":"NOT_NEGATED"}

B.2. Paraphrase Detection Prompt

Your task is to determine if the sentence1 is a paraphrased or negated version of the Sentence2 in Telugu.
Classify the relationship:
- PARAPHRASED: Sentence1 conveys same meaning as Sentence2, by different words or different sentence structure.
- NEGATED: The Sentence1 contradicts or reverses the meaning of the Sentence2.
- If **any part** of the sentence is negated, polarity-flipped, contradicted, or reversed (even partially), you MUST classify it as **NEGATED**.
Given:
Sentence1: "{sentence1}"
Sentence2: "{sentence2}"
Return ONLY a valid JSON object in this exact format:
{"label":"PARAPHRASED"} or {"label":"NEGATED"}
DO NOT include any explanation, text, or formatting outside the JSON.

B.3. Sentiment Classification Prompt

Analyze the sentiment of the following text and classify it as either POSITIVE or NEGATIVE.
Text: "{text}"
Classification Guidelines:
- POSITIVE: Expresses favorable opinions, satisfaction, happiness, praise, benefit, success, approval, progress, desirable situation, or optimistic views
- NEGATIVE: Expresses unfavorable opinions, dissatisfaction, sadness, criticism, harm, failure, danger, loss, undesirable situation, or pessimistic views
- If both positive and negative aspects appear, choose the dominant one.
Return ONLY a valid JSON object in this exact format:
{"sentiment": "POSITIVE"} or {"sentiment": "NEGATIVE"}
DO NOT include any explanation, text, or formatting outside the JSON.

B.4. Polarity Flipping Prompt

In Telugu, polarity is realized on the verb; flipping a verb's polarity can flip the sentence meaning.
If multiple verbs are present, flipping different verbs yields different interpretations, so the target verb or verbs are explicitly provided.
You are given a Telugu sentence where the specified verb(s) contain negative polarity markers.
Rewrite the sentence by flipping the polarity of ONLY the specified verb(s).
Constraints:
- Flip the polarity of the specified verb(s), making only the minimal inflectional adjustments (tense, aspect, mood, person, number, gender) required for grammatical well-formedness
- If the original sentence contains NPIs (e.g., ఏమీ, మరేమీ, ఎవరూ, ఎక్కడికీ, ఏమాత్రం, ఎప్పటికీ), remove or adjust any that are no longer required after the polarity flip.
Input:
Sentence: "{sentence}"
Target Verb(s): "{verb_to_negate}"
Output:
Return ONLY a JSON object in the format:
{"flipped_sentence":"<sentence>"}

C. Negative Polarity Items

Negative Polarity Items (NPIs) are expressions that are grammatically licensed only in the presence of negation. Table 10 presents examples of NPI sentences from the dataset. Mishandling NPIs in downstream tasks can lead to significant errors. In the polarity flipping task, flipping a sentence containing an NPI requires the model to ensure that the NPI remains valid in the resulting sentence, or omit it if it is no longer licensed. In the translation task, sentences with and without NPIs should be translated while preserving their negative meaning.

D. Translation Error Analysis

To better understand how models fail at preserving negation during translation, we conducted a qualitative error analysis on the model outputs. Table 11 presents representative examples of translation errors, categorized by the type of negation failure observed.

E. Additional Results

Table 12 reports detailed accuracy, precision and recall for the sentiment classification task under negated and non-negated instances.

Sentences with Negative Polarity Items (NPI)
నాకు ఏమీ తెలియదని చెప్పింది. *nāku ēmī teliyādani ceppindi* She said that she does not know anything.
గదిలో పాత సామాను తప్ప మరేమీ లేవు. *gadilō pāta sāmānu tappa marēmī lēvu* There is nothing else in the room except old furniture.
శివరావుకి అభిముఖంగా కూర్చున్న సత్యానంద్ ఎప్పుడూ లేనంత ఎక్కువగా తాగుతున్నాడు. *śivarāvuki abhimukhaṃgā kūrcunna satyānaṃd eppuḍū lēnanta ekkuvagā tāgutunnāḍu* Satyanand is drinking more than ever before.
ఆ రొట్టె సరిగా కాలకట్టుంది. అందుకే ఆ రొట్టెను ఎవరూ తినలేదు. *ā roṭṭe sarigā kālakaṭṭuṃdi. aṃdukē ā roṭṭenu evarū tinalēdu* That bread was not properly baked. That is why nobody ate that bread.
నువ్వు ఏం చెప్పొద్దు. *nuvvu ēṃ ceppoddu* You should not say anything.
నేను ఈ విషయం ఎవరికీ చెప్పనుండు. *nēnu ī viṣayaṃ evarikī ceppanuṃḍu* I will not tell this matter to anyone.
ఆకరికి ఎలాంటి సంబోధనా లేకుండానే మొదలెట్టింది. *ākariki elāṃṭi saṃbodhanā lēkuṃḍānē modalleṭṭindi* In the end, she started without any form of address at all.
ఇంతవరకూ ఎవరూ ఎవరికీ ఇవ్వనటువంటి పార్టీ! *iṃtavarakū evarū evarikī ivvanaṭuvaṃṭi pārṭī* A party the likes of which nobody has ever given to anyone!
ఇద్దరూ అలాంటి వాళ్లే అయితే ఇక చెప్పనే అక్కర్లేదు. *iddaru alāṃṭi vāḷlē ayite ika ceppanē akkarlēdu* If both are like that, there is no need to say anything at all.
ఇంత ప్రొద్దున్నే లేవకనా, ఇంటిలో పనంతా చక్కబెట్టింది? *iṃta proddunnē lēvakanā, iṃṭilō panaṃtā cakkapeṭṭindi* Did she finish all the housework without even waking up this early?
అతని మాట ఇంకా పూర్తి కానేలేదు. కన్నారావు కెవ్వుమని అరిచాడు. *atani māṭa iṃkā pūrti kānēlēdu. kannārāvu kevvumani aricāḍu* His words were not even complete. Kannarao screamed.
దేనిమీదా మనస్సు లగ్నం చేయలేక చాలా ఇబ్బంది పడుతుంది. *dēnimīdā manassu lagnaṃ cēyalēka cālā ibbandi paḍutundi* Unable to focus the mind on anything, one is in great difficulty.
తనో చిల్లికానీ పెట్టకుండా ఈ రంగంలోనికి రావాలి. *tanō cillikānī peṭṭakuṃḍā ī raṃgaṃlōniki rāvāli* One must enter this field without spending a single penny.
అడ్రసు కూడా ఇవ్వనంత రహస్యమేముంది దీంట్లో? *aḍrasu kūḍā ivvananta rahasyamēmuṃdi dīṃṭlō* What is so secret that you cannot even give the address?
ఇహ ఇక్కడ ఒక్కక్షణమైనా ఉండకూడదు. *iha ikkaḍa okkakṣaṇamainā uṃḍakūḍadu* One should not stay here even for a single moment.
మళ్ళీ ఓసారి ఒక్కపైసా కూడా ఇవ్వద్దు భరణీ! *maḷḷī ōsāri okkapaisā kūḍā ivvaddu bharaṇī* Do not give even a single penny again, Bharani!

Table 10: NPI sentences in the dataset with NPI words highlighted in blue

Setence	Model Tranlsations	Error
అర్థం కానట్టు అయోమయంగా మళ్ళీ చదివింది. ardhaṃ kānaṭṭu ayōmayaṃgā maḷḷī cadividhi. Confused, she read again as if she didn't understand.	**Bhashaverse:** You haven't read it again. **Indic-gemma-7b:** She read it again to make sure it made sense.	Negation shifted; omitted
బాగా చూస్తేతప్ప అక్కడ చిలుక వున్నట్లు తెలియదు. bāgā cūstētappa akkaḍa ciluka vunnaṭlu teliyadu. Unless you don't observe keenly, you అమ్మ కోపంగా పొద్దస్తమానం టీవీ చూడకని తిట్టింది. amma kōpaṃgā poddastamānaṃ ṭīvī cūḍakani tiṭṭindi. Mom scolded me angrily and told me not to watch TV all the time.	**Llama-3.3-70b:** Mother scolded for watching TV in the morning. **GPT-OSS-20b:** Mom scolded for not watching TV in the morning.	Negation omitted; Mistranslated
ఎక్కువ నీళ్లు తాగకపోతేగూడా, ఇలా వేడి చేస్తుంది. ekkuva nīḷlu tāgakapōtēgūḍā, ilā vēḍi cēstundi. Even if you don't drink much water, it makes you feel this hot.	**Bhashaverse:** Even if you drink a lot of water, it heats up.	Negation omitted
వారే లేనప్పుడు, నేనూ ఉండి ఉండను. vārē lēnappuḍu, nēnū uṇḍi uṇḍanu. When they are not there, I would not be there either.	**Meta-Llama-3-8B:** When they are not there, I will be. **Qwen3-8B:** I will remain here when they are not present.	One negation omitted
వాడు రెండుకంటే ఎక్కువ ఇవ్వడట! vāḍu reṇḍukaṇṭē ekkuva ivvaḍaṭa! He does not give more than two!	**Most of the models:** He gives more than two!	Negation omitted
అనుకున్న పని కాకపోతే మిగిలేది అలాంటి పనికిమాలిన నవ్వే. anukuna pani kākapōtē migilēdi alāṇṭi panikimālin navvē. If the expected work does not happen, what remains is only that useless smile.	**Model:** If the work expected is not done, it's not funny.	Extra negation added
వయసు మీదపడడం వలననో, లేక అనేకమంది శిష్యులలో గుర్తుపట్టకనో అయన మొదట గుర్తుపట్టలేదు. vayasu mīdapaḍaḍaṃ valanano, lēka anēkamandi śiṣyulalō gurtu paṭṭakano ayana modaṭa gurtuppaṭṭalēdu. At first he could not recognize, either because of his advancing age or he has too many disciples.	**Google Translate:** He was not recognized at first, either because of his advancing age or because many of his disciples did not recognize him.	Negation on wrong subject
అందువలన జోగయ్యని చూసీ చూడనట్లు వదిలేసేవాడు. *anduvalna jōgayyani cūsī cūḍanaṭlu vaḍilēsēvāḍu.* Hence he pretended as if he didn't see Jogayya.	**GPT-OSS-20b:** Therefore, Jogayyani would leave as if he had not seen.	Negation on wrong person

Table 11: Examples of negation translation errors made by models, categorized by error type.

	Accuracy		Precision_pos		Recall_pos		Precision_neg		Recall_neg	
Model	NEG	NOT	NEG	NOT	NEG	NOT	NEG	NOT	NEG	NOT
DeepSeek-R1-Qwen3-8B	0.767	0.720	0.923	0.985	0.257	0.536	0.752	0.594	0.991	0.988
Indic-gemma-7b-Navarasa	0.778	0.848	0.618	0.861	0.720	0.883	0.867	0.828	0.804	0.798
Meta-Llama-3-8B-Instruct	0.797	0.857	0.665	0.891	0.695	0.865	0.861	0.811	0.843	0.845
Nemotron-Cascade-8B	0.798	0.744	0.765	0.944	0.502	0.605	0.807	0.621	0.931	0.948
Qwen3-8B	0.833	0.840	0.848	0.962	0.560	0.760	0.829	0.731	0.955	0.957
Gemma-7b-it	0.739	0.758	0.584	0.871	0.537	0.695	0.800	0.656	0.829	0.849
GPT-OSS 120b	0.841	0.870	0.789	0.954	0.662	0.820	0.859	0.782	0.921	0.943
GPT-OSS 20b	0.851	0.873	0.804	0.958	0.685	0.822	0.868	0.785	0.925	0.947
Llama-3.3-70b-versatile	0.867	0.884	0.837	0.970	0.707	0.830	0.878	0.795	0.938	0.962
Gemini-2.5-Flash	0.859	0.890	0.829	0.972	0.685	0.839	0.869	0.804	0.937	0.965

Table 12: Sentiment classification performance under negated (NEG) and non-negated (NOT) sentence conditions.Precision_pos and Recall_pos refer to the positive sentiment class, while Precision_neg and Recall_neg refer to the negative sentiment class.

A Morphological Transducer for the Limbu Language

Avyaya Singh, Jonathan North Washington

Swarthmore College
500 College Avenue, Swarthmore, PA 19081

{asingh5, jwashin1}@swarthmore.edu

Abstract

This paper presents the first ever morphological transducer for the Limbu language, also known by its endonym Yakthung Pan, an endangered Sino-Tibetan language primarily spoken in the area known as Limbuwan/Koshi Province in Eastern Nepal, with a minority population in the Sikkim state of India, where Limbu enjoys official status. Using a corpus of Limbu text produced by field interviews (Michailovsky, 1977), and a translation of the Holy Bible into the Limbu language, the paper presents various elements of the morphology of the language, how they were implemented into the transducer, and an evaluation of the transducer against identified corpora and a gold standard. With a relatively small lexicon, the transducer was found to have reasonable coverage (~60%), with high precision (88%) but low recall (32%). The paper discusses future expansion through further involvement with the community, which can help in the maintenance and revitalisation of the endangered language.

Keywords: Limbu, morphological transducers, morphological analysis, Tibeto-Burman languages, South Asia, Nepal

1. Introduction

This paper describes the creation of the first-ever morphological transducer for the Limbu language of Eastern Nepal, using tools from Apertium, a free and open source digital platform intended primarily for rule-based machine translation.

Linguistic preservation and revitalisation of Limbu is currently being undertaken by groups such as Yakthung Cho, an online Limbu Art collective, run by Subhas Thebe Limbu, a Limbu-origin film director, which seeks to produce Limbu language learning tools through online classes and produce content in the Limbu language aimed at informing the Limbu audience of issues pertaining primarily to their efforts at language revitalisation and autonomist demands in Eastern Nepal.[1] The first author of this paper themselves joined such a class conducted by Yakthung Cho in the English medium to learn the Limbu language.

A morphological transducer, here implemented as a finite-state transducer (FST), both analyses text in the language, providing tokenisation of the text and a morphological analysis of each token (morphological analysis), and creates tokens from input analyses (morphological generation). Morphological transducers can prove helpful in learning vocabulary (Katinskaia et al., 2018) and morphology (Arppe et al., 2022); as elements in word-form creators (Fernald et al., 2016; Kazantseva et al., 2018); and as the basis for spell checkers (Washington et al., 2021). They can further also be used in rule-based and hybrid machine translation systems (Khanna et al., 2021), and in a range of other NLP tasks and pipelines.

This paper is structured as follows. Section 2 provides background into the Limbu language and its general structure. Section 3 describes the methodology of creating the transducer, and has details on encountered challenges. Section 4 presents an evaluation of the transducer, showing its coverage over various corpora as well as precision and recall over a hand-annotated gold standard. Section 5 concludes and discusses future work. Associated tools, including a keyboard and a disambiguator, are also described.

2. Background on Limbu/Yakthung

Limbu, known by its endonym Yakthung Pan (ᤕᤠᤰᤌᤢᤱ ᤐᤠᤴ), is spoken by 350,436 people in Nepal (National Statistics Office, 2025), and by 40,835 speakers in North-Eastern India (Office of the Registrar General and Census Commissioner, 2018), mostly belonging to the Limbu ethnicity.[2] Figure 1 shows the geographic distribution of speakers.

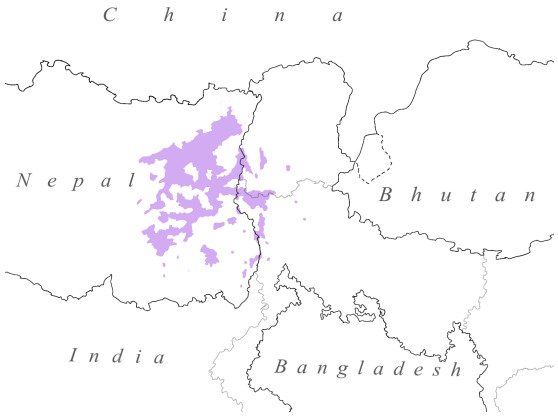


Figure 1 : A map highlighting the regions in which the Limbu language is spoken, namely, Eastern Nepal and the Sikkim state of India.[3]

[1] https://www.instagram.com/yakthungcho/?hl=en

[2] https://www.ethnologue.com/language/lif/

[3] https://commons.wikimedia.org/wiki/File:Limbu.map.png

Limbu belongs to the Kiranti subfamily of the Tibeto-Burman languages within the wider Sino-Tibetan language family. The Kiranti subfamily, spoken in Eastern Nepal includes related languages such as Sunuwar, Rai, and Yakkha. Limbu is spoken primarily in the region known to the Limbu people as Limbuwan, called the Koshi Province by the Nepalese government, and it enjoys official status in the province.[4] Although the province is autonomous, due to the federalisation of the Nepalese government following the republican revolution of 2006,[5] the Limbu people have been demanding the reinstatement of Limbuwan through the renaming of the Koshi province, as these lands are seen as ancestral to them.[6] The language is also spoken by a minority in the Indian state of Sikkim which borders this province to the east, where it also enjoys official status.

Limbu is a highly polysynthetic language: words are inflected using a series of affixes, and multiple words can fuse together. The language has four main dialects: Phedape, Chhathare, Tambarkhole, and Panthare (van Driem, 1987). Interesting linguistic features include dual number, split ergativity, and double (sometimes triple) negation, and primarily Subject-Object-Verb word order. Limbu uses its own indigenous orthography, called the Sirijonga script (§3.3.1).[7]

The phonemic inventory of Limbu is very similar to those of nearby Indo-Aryan languages, although it is itself Sino-Tibetan; this is because it is part of the Nepali Sprachbund. Thus, its phonological inventory and orthography follow the Indo-Aryan *Varga* system, where consonants are divided into 5 *Vargas* according to place of articulation. Thus, in terms of its phoneme inventory, the language and its orthography are comparable to Nepali, Bengali, Hindi, etc. (van Driem, 1987).

Only two reference grammars on the language were available to the authors, both primarily concerning the aforementioned Chhathare dialect: van Driem (1987) and Tumbahang (2007). These grammars were consulted in the creation of the transducer. Because of the more specific dialectal focus of Tumbahang (2007), words/phrases encountered in this source were included as synonyms/alternate spellings of words found in van Driem (2007), in an effort to make the transducer dialect-aware but generalisable for the Limbu language as a whole.

The authors do not speak the Limbu language, but this transducer was created through close study of the aforementioned reference grammars and deductions made through use of linguistic analysis. Future work will include native speakers.

3. Methodology & Implementation

This section goes over the details of the transducer, namely how it was constructed (§3.1), how its lexicon was expanded (§3.2), and solutions to challenges the authors encountered while creating the transducer (§3.3).

3.1 Tools used

The transducer described was created entirely by hand and is freely and publicly available under the GPL v3 Free/Open Source license.[8] The intent is to increase access, as no other transducer is known to exist for the Limbu language, and resources on the language are scarce. The transducer is also available online for morphological analysis and generation via Apertium's web UI.[9] Machine learning approaches were not used, due to their need for large corpora and multiple development cycles, the difficulty in correcting errors, and their lack of precision (Butt, 2021).

The lexd formalism (Swanson and Howell, 2021) was used to implement the morphotactics of Limbu. Previously published lexd transducers showcase its ability to handle non-suffixational morphology (Washington et al., 2021; Christopherson, 2023) and some discontiguous morphology (Swanson and Tyers, 2022), both of which are used extensively in Limbu. The transducer uses the Apertium framework (Forcada et al., 2011; Khanna et al., 2021) for compilation scripts and the HFST format and tools (Lindén et al., 2011) for storing and working with the compiled transducer. Apertium also offers a handful of tools for leveraging a morphological transducer for language maintenance, learning, and revitalisation, which are all future directions the authors intended the transducer for.

The morphophonology was implemented in the twol formalism, included with HFST. The analyser and generator are compiled by compose-intersecting the morphotactics transducer and the morphophonology transducer, as in sources cited above for other lexd transducers.

3.2 Lexicon

The transducer was developed against the only two available digital corpora of Limbu text of any substantial size: the Wycliffe translation of the Bible into the Limbu language (Wycliffe, 2009) and a collection of field interviews in the Limbu language compiled primarily by linguists Boyd

[4] https://www.ethnologue.com/language/lif/

[5] https://interactive.aljazeera.com/aje/2016/nepal-maoist-dream/index.html

[6] https://kathmandupost.com/province-no-1/2023/03/12/general-strike-throws-life-out-of-gear-in-koshi-province

[7] https://www.ethnologue.com/language/lif/

[8] https://github.com/apertium/apertium-lif.git

[9] https://beta.apertium.org/#analysis?aLang=lif

Michailovsky and Martine Mazaudon in 1977 (Michailovsky, 1977), available on the Pangloss website.[10,11] This latter source is not in the official Indigenous script of the language, and thus parts of it were hand-transliterated by the first author for the purposes of evaluating this transducer. The first author used a dictionary app in Limbu, available online, which contains basic words from every dialect, as a primary source for specific words in the language.[12] Besides these primary sources, the first author also used an online database[13] and dictionary[14] of Limbu words, both presented in the Latin transcription.

Several methods were used to expand the lexicon. One involved examining the most frequently used words in the Limbu Bible translation and cross-referencing them with an English translation. Through this method, many verbs and proper nouns were able to be added (such as God, David, Solomon, Earth, Heaven; saying, doing, commanding). A similar method was used in relation to parts of the Pangloss corpus (Michailovsky, 1977), focussing on the *Pandy Dancing - A Conversation* interview,[15] one of the longest of the collection of interviews in this source. Further, the first author used the aforementioned Dictionary app and reference grammars to check noun stems and other parts of speech. Through an iterative development cycle (Tyers et. al, 2015) using these resources, the first author was able to generate a basic lexicon for Limbu, made up of the most common words encountered in the corpora (Table 1).

Parts of Speech	Tag(s)	Stems
Numerals	<num>	157
Nouns	<n>	148
Proper Nouns	<np>	131
Pronouns	<prn>	56
Verbs	<v>	51
Adverbs	<adv>	38
Ordinals	<num><ord>	22
Adjectives	<adj>	17
Prepositions	<pr>	10
Conjunctions	<cnj>	10
Determiners	<det>	7
Interjections	<ij>	5
Others		52
Total		**704**

Table 1: Lexicon size by part of speech

[10] https://pangloss.cnrs.fr/corpus/Limbu

[11] These sources are not entirely ideal, but little else is available. The work was partly informed by the Limbu classes the first author joined. A Limbu Wikipedia (https://incubator.wikimedia.org/wiki/Wp/lif/Main_Page) has been added to Wikimedia Incubator, but as of yet, it only contains a few sentences in the language. A translation of the Universal Declaration of Human Rights also exists, available only as an image.

[12] https://play.google.com/store/apps/details?id=com.jettyland.limbudictionary

[13] https://starlingdb.org/cgi-bin/query.cgi?basename=%2fdata%2fsintib%2flimet

[14] https://pangloss.cnrs.fr/dictionaries_content/limbu/dico.html, compiled by Boyd Mikhailovsky

[15] https://doi.org/10.24397/pangloss-0004182

The transducer consists of 125 tags, mostly following the Apertium tagset,[16] defined in the lexd formalism as multi-character symbols, between less-than '<' and greater-than '>' characters. 11 of these tags cover the basic parts of speech as described in Table 1; the rest correspond to subcategories (e.g., noun class and pronoun type), morphological functions (e.g., number, negation marking, person marking, etc.), and miscellaneous POS-level tags (e.g., emphasis and question markers and punctuation tags).

3.3 Challenges

This section discusses a number of challenges encountered in the creation of the transducer: the Yakthung indigenous orthography (§3.3.1), discontinuous verbal morphology (§3.3.2), and morphological ambiguity (§3.3.3).

3.3.1 Orthography

The Limbu language uses an Indigenous orthography called the Sirijonga script, which was purportedly designed by Limbu King Sirijanga Hang in the 9th century, and was then revised by Te-ongsi Sirijunga Xin Thebe in the 18th century, who, due to his work on the Limbu language, came to be considered the reincarnation of the former. No text from before the 18th century exists, and thus, it is believed that Xin Thebe actually designed the script himself (Gaenszle, 2021). An abugida which descends from the wider Brahmi family of scripts used in the region, it is supported by Unicode[17] (original proposal: Michailovsky, 2002). An example of the script is provided in Figure 2.



Figure 2: A sentence in the Sirijonga script of the Limbu language, *yakhtuŋ pan chigemasum*, translating to "You've lost a little of the Limbu language" (Michailovsky, 1977).[18]

Because Sirijonga is an abugida where vowels are diacritics and there are many morphophonological alternations, it was often more straightforward to encode segments as multicharacter non-Sirijonga symbols that could be resolved to Sirijonga based on context. An example of this is the verb 'to do' (Figure 3), encoded entirely with multicharacter symbols. The twol rule in Figure 4 ensures that the {Kk} symbol is realised as ◌ᤴ [k] or ᤺ [g] depending on phonological context. Examples of resulting forms are shown in Figure 5.



Figure 3: Encoding of the verb *chōkma* 'to do' in the lexd formalism using multicharacter symbols.

[16] https://wiki.apertium.org/wiki/List_of_symbols

[17] https://www.unicode.org/charts/PDF/U1900.pdf

[18] *Paddy-dancing - a conversation*, Sentence 292. https://doi.org/10.24397/pangloss-0004182#S292

-3	0	+2	+4	+7	+8	+9	+12
ᰀᰤ- *a-* SUBJ.1PL.INCL-	ᰀᰤᰎ *ab* shoot	**-Ø** -NPST	- ᰥ *-u* -OBJ.3	- ᰬᰮ *-m* -SUBJ.1PL	-ᰦ *-si* -SUBJ.NSG	- ᰬᰮ *-m* -SUBJ.1PL	**-Ø** -PERF
ᰅ- *ke-* SUBJ.2PL-	ᰠᰵᰟ *baʔr-* speak	**-Ø** -NPST	- ᰤ *-i* -OBJ/SUBJ.PL		**-Ø** -OBJ.SG		**-Ø** -PERF

Table 2: Morphological breakdown of two example verb forms: (a) ᰀᰀᰥᰬᰦᰮ *aabumsim* ‘We (incl) have shot them’ and (b) ᰅᰠᰵᰟ *kebaʔri* ‘You (pl) speak (obj)’. **Ø** indicates an empty slot corresponding to a default.

"{Kk} becomes ᰎ if succeeded by a vowel"
%{Kk%}:ᰎ <=> _ %>: (:ᰥ) (:0) :Vowel ;

Figure 4: Twol rule for the {Kk} symbol.

ᰃᰬᰅᰟ:ᰃᰬᰅᰟ<vbdo><inf>
ᰃᰬᰎᰟ:ᰃᰬᰅᰟ<vbdo><nonpast><s_sg3><o_3><o_sg><perf>

Figure 5: Example forms analysed by the transducer involving the twol rule from Figure 2, with a [k] in the first example a [g] in the second. The ‘:’ separates forms from analyses.

In order to fully implement the morphophonological alternations of the language, 45 twol rules were encoded.

In addition to this project, the first author also created a keyboard for the Sirijonga script, as currently available keyboards were not fully intuitive, as cited by members of the community (p.c., February, 2025). Existing keyboards were based on QWERTY,[19] created by a non-Limbu Nepali artist,[20] or based on the consonant *Varga* system (available in Google's Gboard[21]). The keyboard, developed based on glyph-frequency, was made in consultation with members of the Limbu community, who commented on its effectiveness, and it is available under the MIT License on GitHub as a free download.[22] It was first created using iBus-m17n,[23] and documentation of the layout is also available.[24] In order for seamless usage across platforms, the first author worked with Translations Commons, a language digitisation NGO, on making a mobile-friendly version of the author’s produced keyboard. This Free/Open Source keyboard, made using the Keyman system,[25] is supported on iPadOS, Android, iOS, and Windows, and is available for free distribution by the author (60 downloads over the last 2 months) and on the Translations Commons website,[26] as well as on the official Keyman website.

3.3.2 Verbal Morphology

Being a highly polysynthetic language, Limbu can agglutinate up to ~15 suffixes and ~5 prefixes (including cliticised adverbs) to a single verb root, and also incorporate other parts of speech into a verb form (van Driem, 1987). There are three negation affixes that can appear in Limbu also verbal morphology, amounting to the same meaning (the language does not possess double negation as a form of emphasis). For each verb, no affix is mandatory, and thus, a root can exist independently, or with the addition of up to all ~20 affixes. The absence of a filled affixal slot can either mean ‘default’ for that particular slot (singular for number, first person for person, active for voice, etc.) or simply be an open slot, not providing any additional meaning to the verb. Example verb forms are shown in Table 2, and one pattern for verbal morphology included in the transducer is shown in Figure 6.

Adverbs? + SPN(1) + SPN(4) + Neg?(1) + **Verb-Stems** + Neg?(2) + Reflexivity? + Tense + SPN(5) + SPN(2) + ObjSlot + SPN(6) + NegS? + SPN(7) + ObjNum + SPN(8) + SPN(3) + NegS2? + Mode? + Aspect + Clitic? + Nom? + Subord? + *Adverbs?*

Figure 6: One verbal morphology pattern in the transducer, slightly simplified. *Italics* (added for demonstration) represent lexicons containing foreign roots, **Bold** (added for demonstration) represents the lexicon containing verbal roots, SPN represents the lexicon containing affixes related to the subject’s person and number, Neg represents negation lexicons. Numbers in parentheses represent lexicon columns. Some columns include only tags that match prefix morphemes.

Although 13 other such patterns for verbs exist, as verbal morphology varies greatly in the language, the patterns are precise, meaning an error cannot occur in identifying which pattern a verb follows (and thus which slot an affix falls in). Patterns were identified based on van Driem (1987), Michailovsky (1977), and linguistic analysis on forms not encountered in these sources. The morphology of nouns is only somewhat less complex, with 7 patterns being

[19] https://keyman.com/keyboards/sil_limbu_phonetic
[20] https://keyman.com/keyboards/sil_limbu_typewriter
[21] https://support.google.com/gboard/answer/6380730#zippy=%2Cfind-supported-languages
[22] https://github.com/SwatLangTech/Limbu_Keyboard
[23] https://github.com/ibus/ibus-m17n
[24] https://wikis.swarthmore.edu/ling073/Limbu/Keyboard
[25] https://keyman.com/
[26] https://reco.to/JZazvh, https://translationcommons.org/keyboard-creation-projects/published-keyboards/

included. The patterns included so far in the transducer cover the most common morphological patterns in the language.

3.3.3 Ambiguity

The aforementioned 15+5 affixes have many variants, due to phonological alternations, dialectal variation, and the fact that there appear to simply be synonymous morphemes. This is compounded by the fact that there exist 13-15 noun cases (van Driem, 1987) in this language, with each having between 2-5 lexical variants, with the genitive case having the most variants, at 11 (Figure 7), attributed to both orthographic variation and dialectical differences in the language. In the lexd file, the first line of every block of synonymous morphemes is used to denote the 'standard form', selected based on the reference grammars. The subsequent variants are either dialectical variants as denoted by Tumbahang (2007) or lexical variants encountered in the Pangloss collection of Limbu interviews. All forms besides the "standard form" include Dir/LR in the comment, which allows them to be included in the analysis transducer but excluded from the generation transducer.[27] Currently, the case lexicon itself contains 51 lines of code in order to account for these variations, 36 of which are considered variants.

```
<gen>:>{NG}{A}{n}
<gen>:>ᤛ                # Dir/LR
<gen>:>ᤜ                # Dir/LR
<gen>:>{l}ᤜ             # Dir/LR
<gen>:>{ie}{l}ᤜ         # Dir/LR
<gen>:>{a}{ie}{l}ᤜ      # Dir/LR
<gen>:>ᤝᤰ               # Dir/LR
<gen>:>ᤔᤰ               # Dir/LR
<gen>:>{l}ᤔᤰ            # Dir/LR
<gen>:>{ie}{l}ᤔᤰ        # Dir/LR
<gen>:>{a}{ie}{l}ᤔᤰ     # Dir/LR
```

Figure 7: Variants of the genitive case based on dialectical or orthographical variation.

Additionally, some morphemes appear to be syncretic with other morphemes, despite being rarely used. For example, -/re/ is one of the alternative genitive case markers, but -/re/ can also mark vocative and ergative. Hence any token containing -/re/ has at least 3 analyses.

A disambiguator was created in VISL CG-3 (Bick and Didriksen, 2015) to reduce the ambiguity of analyses by context. An excerpt is provided in Figure 8—one heuristic which deals with the ambiguity of the genitive suffix, with an example of its application in Figure 9.

```
"<ᤖᤛ>"
        "ᤖ" n sg gen SELECT:85
;       "ᤖ" n sg erg SELECT:85
;       "ᤖ" n sg voc SELECT:85
"<ᤀᤛᤧ>"
        "ᤀᤛᤧ" n sg
        "ᤀᤛᤧ" n ser3sg
;       "ᤀᤛᤧ" n attr REMOVE:293
```

[27] See Washington et al. (2021, 187) for details on this approach.

Figure 9: Disambiguation of forms in context ('wheat-GEN bread') employing the rule in Figure 8 ("85"). Analyses with preceding ; are removed by the rules from the list of possible analyses.

```
LIST NOUN = n np ;
LIST NGEN = (n gen) (np gen) ;

SELECT NGEN IF
  (0 NGEN)
  (1 NOUN) ;
```

Figure 8: A CG-3 rule from the disambiguator.

The Limbu disambiguator (apertium-lif.lif.rlx) consists of 88 rules. The performance of the disambiguator is discussed in §4.1.

4. Evaluation

For the sake of evaluating this transducer, several corpora were prepared using the previously described only extant digital corpora of Limbu text in the Sirijonga orthography: the Bible corpus. Genesis 1 & 2 as well as 10 sentences (of 324) from the *Paddy Dancing* interview were used for development and are hence treated separately.

4.1 Naive Coverage and Ambiguity

The transducer[28] was evaluated in terms of naïve coverage, which is defined as the number of forms in a corpus which return an analysis, irrespective of whether this analysis is correct or not. The ambiguity of the transducer on these corpora, defined as the average number of analyses returned by the analyser per analysed token, was also calculated with and without the disambiguator.[29] The results are in Table 2.

corpus	tokens	coverage	ambiguity before / after
10 sents	177	100.00 %	3.82 / 1.62
Gen. 1 & 2	1253	63.05 %	3.03 / 1.69
Full Bible	759K	60.62 %	2.28 / 1.44

Table 2: Naive coverage and ambiguity on development and test corpora.

It's seen that coverage is reasonable for a young transducer, at over 60%. Ambiguity is reduced greatly by the disambiguator, from ~3 analyses per token to around 1.5.

4.2 Accuracy

Precision and recall were measured to understand the accuracy of the transducer. Precision is defined as the percentage of analyses returned by the transducer for each form that are correct. Recall is defined as the percentage of the correct analyses for each form that are returned as opposed to not being returned at all.

[28] The transducer was last evaluated on revision 4b0ebbb in March, 2026.

[29] Scripts used for calculating coverage, ambiguity, precision, and recall are available in the `dev/` directory in the transducer's git repository.

In order to measure accuracy, the first author randomly selected 100 unique tokens (of ~4k) from the *Paddy Dancing* interview, hand-transliterated them, and annotated them manually to create a gold standard. The words in the gold standard appeared to be generally representative of the lexicon and morphology of texts the authors have worked with. The gold standard attempted to capture all possible analyses for each token, by using the glosses provided by Mikhailovsky (1977) and leveraging the first author's knowledge of Limbu morphology. The gold standard was compared to the list of analyses returned by the analyser.

Precision was measured at ~88% (108/122 analyses), while recall was only ~32% (108/340 analyses), with an F-score of 0.4675; i.e., most analyses returned by the transducer were correct (except in two instances, noted below), but many correct analyses were not returned due to the fact that the transducer has low coverage on the interview.

A preliminary qualitative evaluation of forms that did not have an analysis returned by the transducer yields several patterns. Besides missing lexical items (including verb stems, adverbs, nouns, classifiers, etc.), there were a few uncommon morphotactic patterns not implemented in the transducer, mostly for verbs, such as polar question and reduplicated forms. Additionally, there are some mistakes in pattern definitions that cause some patterns not to function fully as intended; e.g., while the language allows up to three negation morphemes in a verb, and the morphotactics were designed to support this, currently the transducer only recognises up to two negation morphemes in a verb form.

Given the fact that the majority of analyses not returned by the transducer were caused by stems missing from the lexicon, the low recall result substantiates the fact that the stems in the lexicon are indeed more common than those missing. Assuming similar numbers of possible analyses per form, and given the random unique sampling approach used to build the gold standard, if included and missing stems were evenly distributed in frequency, then recall would approximate coverage.

There were very few forms with incorrect analyses returned from the transducer; all involve hypothetical but unattested verb forms which are synonymous with another word. For example, the word ᰕᰤᰪ *char* refers to the numeral “4”, however, it is also the predicted verb form involving the optative suffix -ᰪ *-r* added to the verb ᰕᰤᰖ *chama*. However, ᰕᰤᰪ *char* is always used to reference the numeral, and an alternative optative form, ᰕᰤᰍ *chal*, is instead used for ᰕᰤᰖ *chama*.[30] Thus, the optative analyses returned by the transducer in this instance are incorrect. Similarly, the word ᰍᰥᰤ *waʔi* refers to a polar question verb form of the stem ᰍᰖ *wama* "to be" in all contexts, rather than any of the other analyses returned by the transducer, despite these being predicted by the morphotactics. One of these analyses involves object marking, which is currently incorrectly allowed for verbs which are not transitive.

[30] For other verbs, the two optative forms are used interchangeably.

5. Conclusion & Future Work

This paper presents the first ever morphological transducer for the endangered Limbu language, to the best of the authors' knowledge. With significant coverage on the Limbu Bible, and with high precision, this paper contributes to the linguistic description of the language, and introduces the ability to perform various text-processing tasks in the language, enhancing the digital presence of the language.

In terms of future work on the language, the authors believe that coverage can be expanded in two main ways: (1) by hand-transcribing Limbu print books, the Universal Declaration of Human Rights, and the interviews from Mikhailovsky (1977), automating transliteration of other sources, and expanding the lexicon based on their contents, and (2) working more closely with native speakers to make sense of forms encountered in these sources that are not able to be understood using published resources. The transducer is easily scalable through adding to the lexicon, tweaking the existing morphological patterns, and adding new ones.

As mentioned before, this transducer can also be used as a component in a rule-based machine translation system. Given that there are currently no MT systems for the Limbu language, the first author did create a basic Limbu-English MT system as a proof-of-concept. However, it is still quite barebones, and is only able to handle a few simple phrase types and words of the language. The authors seek to expand this MT system in the future.

This transducer and its associated projects (e.g., the Limbu keyboard) have already been introduced to the Limbu community. There appears to be some initial uptake, and community members have commented on the efficiency and usefulness of these projects.

Acknowledgments

The authors would like to thank Subhas Thebe Limbu[31,32] and Sumi Limbu[33] for their support and for their help in looking over our work. We would also like to thank all members of the Limbu community for keeping this language alive and thriving, and Yakthung Cho for their contribution

[31] subashthebe.com/

[32] eyebeam.org/artists/subash-thebe-limbu/

[33] scholarworks.uark.edu/etd/5325/

to Limbu linguistic preservation through online classes.[34]

References

Arppe, Antti & Poulin, Jolene & Neitsch, Andrew & Harrigan, Atticus & Hieber, Daniel (2022). itwêwina: Towards a morphologically intelligent and user-friendly online dictionary of Plains Cree. 10.13140/RG.2.2.33073.08804.

Bick, Eckhard & Didriksen, Tino. (2015). CG-3 - Beyond Classical Constraint Grammar. Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015), Vilnius, Lithuania.

Butt, Miriam (2020). Building Resources: Language Comparison and Analysis. At the Second Workshop on Computational Research in Linguistic Typology (SIGTYP 2020), invited to talk. https://youtu.be/D2AzyAY_3Mc.

Christopherson, Cody Scott (2023). A finite-state morphological analyzer for Q'eqchi' using Helsinki Finite-State Technology (HFST) and the Giellatekno infrastructure. Master's thesis, Brigham Young University.

van Driem, George (1987). A Grammar of Limbu, Berlin, New York: De Gruyter Mouton. https://doi.org/10.1515/9783110846812

Fernald, Theodore B., Kashyap, Nabil, and Fahringer, Jeremy (2016). Navajo verb generator. Development version.

Forcada, Mikel L., Ginestí-Rosell, Mireia, Nordfalk, Jacob, O'Regan Jim, Ortiz-Rojas, Sergio, Pérez-Ortiz, Juan Antonio, Sánchez-Martínez, Felipe, Ramírez-Sánchez, Gema, and Tyers, Francis M. (2011). Apertium: a free/open-source platform for rule-based machine translation. Machine Translation, 25(2):127–144.

Gaenszle, Martin (2021). The Limbu Script and the Production of Religious Books in Nepal. Philological Encounters. 6 (1–2): 43–69. doi:10.1163/24519197-bja10014. ISSN 2451-9197.

Katinskaia, A., Nouri, J., and Yangarber, R. (2018). Revita: a language-learning platform at the intersection of ITS and CALL. In Proceedings of LREC: 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan.

Kazantseva, Anna, Maracle, Owennatekha Brian, Maracle, Ronkwe'tiyóhstha Josiah, and Pine, Aidan (2018). Kawennón:nis: the wordmaker for Kanyen'kéha. In Proceedings of the Workshop on Computational Mod- eling of Polysynthetic Languages, pages 53–64, Santa Fe, New Mexico, USA. Association for Computational Linguistics

Khanna, T., Washington, J., Tyers, Francis M., Bayatlı, Sevilay, Swanson, Daniel G., Pirinen, Tommi A., Tang, Irene, and Alòs i Font, Hèctor (2021). Recent advances in Apertium, a free / open-source rule-based machine translation platform for lowresource languages. Machine Translation.

Lindén, K., Silfverberg, Miikka, Axelson, Erik, Hardwick, Sam, and Pirinen, Tommi (2011). HFST— Framework for Compiling and Applying Morphologies, volume 100 of Communications in Computer and Information Science, pages 67–85. Springer.

Michailovsky, Boyd; Everson, Michael (2002). L2/02-055: Revised proposal to encode the Limbu script in the UCS (PDF). Retrieved 2026.

Michailovsky, Boyd, Mazaudon Martine (1977). Pangloss Collection | Limbu Corpus. URL: https://pangloss.cnrs.fr/corpus/Limbu?lang=en&mode=norma

National Statistics Office (2025). Languages in Nepal. Kathmandu: National Statistics Office. (National Population and Housing Census 2021). URL: https://censusnepal.cbs.gov.np/results/files/result-folder/Language%20in%20Nepal.pdf

Office of the Registrar General & Census Commissioner (2018). Language: India, States and Union Territories (C-16_25062018). In Census of India 2011. URL: https://www.censusindia.gov.in/nada/index.php/catalog/42458

Swanson, Daniel and Howell Nick (2021). Lexd: A finite-state lexicon compiler for non-suffixational morphologies In Mika Hämäläinen, et al., editors, *Multilingual Facilitation*, pages 133–146. Helsingin yliopisto.

Swanson, Daniel G. and Francis M. Tyers (2022). Handling Stress in Finite-State Morphological Analyzers for Ancient Greek and Ancient Hebrew. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 108–113, Marseille, France. European Language Resources Association. https://aclanthology.org/2022.lt4hala-1.15/

Tumbahang, Govinda Bahadur (2007). A Descriptive Grammar Of Chhatthare Limbu.

[34] https://docs.google.com/forms/d/e/1FAIpQLSctveBqsNhZIIZgUFA-4cneHCV1ghaDCOIIBGCBdN04EzrDBg/viewform

Kathmandu: Central Department of English Kirtipur. hdl:123456789/3017.

Tumbahang, Govinda Bahadur (2007). A Descriptive Grammar Of Chhatthare Limbu. Kathmandu: Central Department of English Kirtipur. hdl:123456789/3017.

Tyers, Francis, Pirinen, Tommi, Washington, Jonathan (2015) Finite-state morphologies and text corpora as resources for improving morphological descriptions. *Workshop on Computational Phonology and Morphology.* University of Chicago. Poster presentation.

Washington Jonathan, Lopez, Felipe, and Lillehaugen, Brook (2021). Towards a morphological transducer and orthography converter for Western Tlacolula Valley Zapotec. In Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas, pages 185–193, Online. Association for Computational Linguistics.

Washington, Jonathan, Salimzianov, Ilnar, Tyers, Francis M., Gökırmak, Memduh, Ivanova, Sardana, and Kuyrukçu, Oğuzhan (2021). Free/open-source technologies for Turkic languages developed in the Apertium project. In Proceedings of the Seventh International Conference on Computer Processing of Turkic Languages (TurkLang 2019).

Wycliffe Bible Translators, Inc. & Isai Limbu Literature Association (2009). Limbu New Testament (Limbu Script). Digital Bible Society. URL: https://www.scriptureearth.org/00eng.php?idx=1480&LN=Limbu. Accessed: February, 2026. Currently only available as a PDF: https://find.bible/bibles/LIFWBT/

Evaluating Large Language Models for Medical Named Entity Recognition in Urdu: A Benchmark Study

Bushra Nasim[1], Kinza Latif[1], Muhammad Zohair[1], Muhammad Hassan Asif[1], Zarmeen Nasim[2]

[1] Dow Medical College, Karachi, Pakistan, [2] CITRIC Health Data Science Centre, Aga Khan University, Karachi, Pakistan
{bushra.nasim02, kinzalatif01, zohair.ajaz, muhammadhassanasif02}@gmail.com, zarmeen.nasim@aku.edu

Abstract

Medical named entity recognition (NER) is a crucial task in natural language processing (NLP) for extracting meaningful entities such as diseases, symptoms, medications, body parts, and treatments from clinical text. However, NER in low-resource languages like Urdu remains underexplored due to limited annotated datasets. In this study, we evaluated the performance of two state-of-the-art large language models (LLMs), ChatGPT-4o and LLAMA 3.2, on Urdu medical NER using a dataset of 2,057 health-related Urdu news headlines manually annotated across five entity categories. Both models were evaluated using precision, recall, and F1-score. It was found that both models exhibited low precision and moderate recall. ChatGPT-4o achieved the highest F1 for Disease (0.35) while LLAMA 3.2 reached slightly lower F1 scores for Disease (0.33). Both models performed poorly on treatment-related terms, with F1 scores of 0.036 (LLAMA 3.2) and 0.011 (ChatGPT-4o). Micro-average F1-scores were 0.187 for ChatGPT-4o and 0.183 for LLAMA 3.2, indicating comparable overall performance. These findings highlight the challenges of medical NER in low-resource languages and underscore the need for domain-specific fine-tuning, transfer learning, few-shot learning, and prompt engineering to improve performance.

Keywords: medical NER, low resource language, Urdu language processing, LLM, healthcare, medical entities

1. Introduction

In natural language processing (NLP), Named Entity Recognition (NER) is a fundamental task aimed at determining and categorizing entities within the given text. The task is crucial for various domains including healthcare. In the context of healthcare domain, efficient and accurate labelling of named entities enables a wide range of applications, including clinical record summarization, automated extraction of symptoms and diagnoses, pharmacovigilance, and public health surveillance. Recent advances in large language models (LLMs) have made significant improvements across various NLP tasks such as sentiment analysis(Deng et al., 2023), machine translation(Coleman et al., 2024; Elshin et al., 2024), NER(Iwakami et al., 2024), text generation(Kumichev et al., 2024) high-resource languages. Nonetheless, the applicability of these models to low-resource languages remains underexplored. Most LLMs are pre-trained predominantly on English and other high resources languages, resulting in performance disparities when applied to low resource languages.

Low-resource languages are often characterized by limited digital data, annotated resources and preprocessing tools. Urdu, a national language of Pakistan and spoken by 300 million people (Nasim & Haider, 2022) worldwide, is considered a low resource language with limited labeled data available specifically for medical domains. Medical NER in Urdu presents additional linguistic and technical challenges. These include rich morphology, free word order, absence of capitalization cues for entity boundaries and dialectal variation. Such characteristics complicate entity recognition and reduce the effectiveness of models trained primarily in high-resource languages.

Recent studies have explored the capability of large language models (LLMs) in English language for medical NER, reporting promising results in zero-shot or few-shot settings (Jahan et al., 2024). However, their effectiveness in specialized domains and low-resource settings like Urdu medical text remains unclear. Systematic benchmarking in this context is therefore necessary to understand both their potential and their limitations.

In this study, we aim to evaluate the performance of the two state-of-the-art LLMs: ChatGPT-4o and LLAMA 3.2 on medical NER task in Urdu language. We focus on identifying five critical entity categories: diseases, symptoms, medications, treatments, and body parts. These entities are central to healthcare text analytics and have direct implications for clinical decision support and public health monitoring. To facilitate this evaluation, we created a manually annotated dataset of 2,057 Urdu health-related news headlines. The primary objective of this work is to benchmark the zero-shot performance of general-purpose LLMs on Urdu medical NER and to analyze the challenges encountered in this low-resource, morphologically rich language setting.

The major contributions of this study are as follows:

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 40–48
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

- To the best of our knowledge, we present the first manually annotated dataset specifically designed for medical NER in Urdu.
- We provide systematic benchmarking of two state-of-the-art LLMs on Urdu medical NER, offering quantitative and qualitative insights into their strengths and limitations.
- We analyze the linguistic and domain-specific challenges that affect LLM performance in a low-resource South Asian language, highlighting implications for future research in healthcare NLP.

2. Related Work

Named entity recognition (NER) is an important domain of Natural Language Processing (NLP) with increasing applications in the medical domain. Various methodologies have been employed for medical NER, ranging from traditional rule-based systems to advanced deep learning approaches. While significant progress has been made in high-resource languages like English, the development of medical NER in low-resource languages such as Urdu, remains underexplored. This section reviews advancements in medical NER, focusing on the challenges in low-resource languages and the emerging role of LLMs in addressing these challenges.

In high-resource languages, medical NER has benefited significantly from advances in deep learning and pre-trained models. Models like BERT-based BioBERT(Lee et al., 2020a) and ClinicalBERT (Alsentzer et al., 2019), trained on biomedical corpora, have achieved state-of-the-art performance by leveraging domain-specific embeddings to capture complex medical terminology (Lee et al., 2020b). Other approaches, such as Conditional Random Field (CRF) (Zhao et al., 2012) and BiLSTM-CRF architectures (Nasim et al., 2020) , have been used for sequence labeling tasks and remain popular due to their ability to incorporate contextual dependencies (Lample et al., 2016). Despite these advances, direct application of such models to low-resource languages like Urdu is limited by the lack of annotated datasets and domain-specific pre-training.

LLMs such as GPT-3 (Zhang & Li, 2021), PaLM(Anil et al., 2023), and ChatGPT(Wu et al., 2023) have demonstrated remarkable performance in general NLP tasks, including NER (Brown et al., 2020). These models are trained on massive multilingual datasets, which allow them to generalize across languages and domains. Efforts to adapt LLMs for NER include techniques like LoRA (Low-Rank Adaptation), which fine-tunes pre-trained models efficiently for domain-specific tasks (Wang et al., 2023). While these methods show promise, integrating the generative outputs of LLMs into structured tasks such as NER remains an ongoing challenge, particularly in low-resource settings.

NER in low-resource languages such as Urdu faces unique challenges, such as lack of capitalization, free word order, morphological richness, a context-dependent lexicon, and the scarcity of training data(Naz et al., 2014). Most of the work on Urdu NER has focused on general domains rather than specialized fields like medicine. For example, Anam et al. (Anam et al., 2024) proposed using Floret embeddings combined with BiLSTM, GRU, and CRF for general Urdu NER.

However, these models often fall short in domain-specific tasks like medical NER due to the lack of specialized training data in the target language. Medical texts are inherently complex, containing domain-specific terminologies, abbreviations, lack of formatting, missing punctuations, misspelling that require a nuanced understanding of the language and its medical semantics(Leaman et al., 2015). The challenge is amplified in low-resource languages like Urdu, where limited annotated datasets and pretraining corpora are available for domain adaptation. Researchers have explored hybrid methods, such as combining rule-based approaches with machine learning techniques, but these approaches are less adaptable to the complex structure of medical text (Ahmed et al., 2024).

Research in NER for Urdu, a low-resource language, has largely focused on general domains. Recent studies have highlighted the limitations of these approaches in capturing domain-specific context, especially for medical NER, where abbreviations, synonyms, and code-switching are common. Furthermore, the lack of annotated datasets in Urdu for medical text is a critical challenge. High-quality labeled datasets are required for training and evaluating advanced deep learning models. Nonetheless, developing such datasets is labor-intensive and requires domain expertise. For Urdu, the scarcity of annotated corpora in the medical domain significantly hinders progress. Unlike high-resource languages such as English, where extensive annotated datasets exist for medical text, Urdu suffers from a lack of standardized resources.

To the best of our knowledge, this study is the first attempt of evaluating the performance of LLMs for medical NER in Urdu language, highlighting a significant gap in the existing literature. Additionally, our study seeks to fill the gap of scarce resources by introducing the first annotated dataset for Urdu medical NER. This contribution is expected to bridge the gap between NLP and healthcare in Urdu language, providing data and insights that would have a significant impact on clinical text processing.

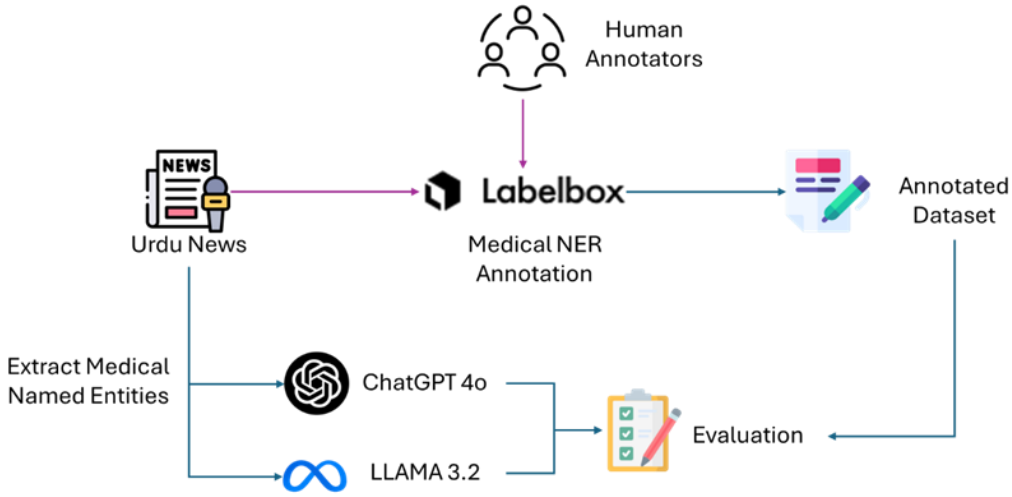


Figure 1: Workflow of Evaluating LLMs on Urdu Medical NER Task

3. Methodology

This section describes the dataset used and the large language models (LLMs) evaluated in this study for the medical named entity recognition (NER) task. We used a publicly available dataset of Urdu news[1], which includes news headlines and summaries. The subset of the complete dataset comprising of 2057 Urdu news headlines was carefully annotated for five medical named entities: *Disease, Symptom, Treatment, Medication, and Body Part*. The dataset was used to evaluate the performance of LLMs in extracting medical entities from the given Urdu text. Figure 1 illustrates the workflow of the methodology.

3.1 Dataset Preparation

The dataset of 2057 news summaries were annotated manually using LabelBox[2]. Four human annotators with educational background in medical sciences were recruited for NER task. The annotators were given a demonstration of the tool and were provided with the definition of each entity tag. The first 1000 news summaries were labeled independently by two annotators. F1-Score and percentage of exact agreement was calculated to assess the Inter-annotator agreement, yielding scores of **0.91** and **90.56%,** respectively. For the final entity labels, the discrepancies and conflicts were resolved through group discussions among the annotators and the research team. After confirming consistent annotation quality, the remaining 1057 headlines were labeled by a single annotator. Figure 2 presents a few examples of annotated news summaries.

3.2 LLMs Evaluated

In this study, we evaluated two large language models (LLMs) for the task of medical named entity recognition (NER) in Urdu: ChatGPT 4o and

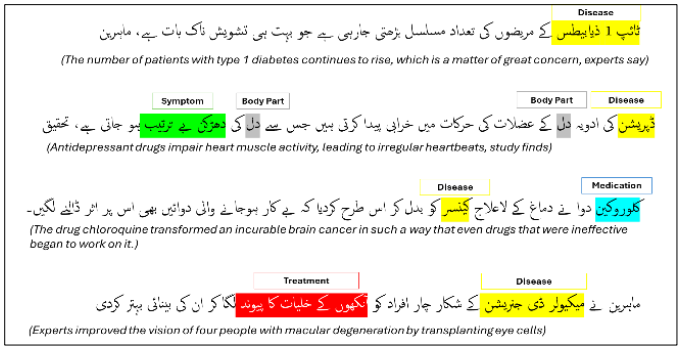


Figure 2: Instances of Annotated News Summaries

LLaMA 3.2. These models were selected due to their multilingual capabilities, accessibility, and increasing adoption in NLP research. Both models were evaluated in a zero-shot setting without task-specific fine-tuning.

3.2.1 ChatGPT-4o

ChatGPT-4o, developed by OpenAI, is a large-scale multimodal language model trained on diverse web-scale corpora. The model supports multiple languages, including Urdu. In this study, ChatGPT 4o was accessed through its web interface and prompted directly for entity extraction without additional fine-tuning or parameter adaptation. The evaluation therefore reflects its zero-shot performance on Urdu medical text.

3.2.2 LLaMA 3.2

LLaMA 3.2, developed by Meta, is an open-weight large language model designed for multilingual and instruction-following tasks. Similar to ChatGPT 4o, LLaMA 3.2 was evaluated in a zero-

1 https://github.com/mwaseemrandhawa/Urdu-News-Headline-Dataset

2 https://labelbox.com/

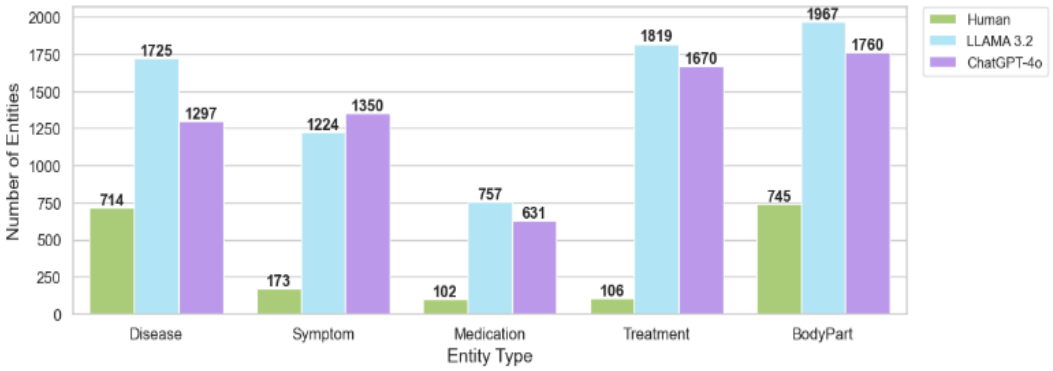


Figure 3: Entities Extracted by LLMs vs Humans

shot setting using its web-based interface. No domain-specific adaptation, fine-tuning, or parameter-efficient techniques were applied.

3.3 Prompt Design and Evaluation Procedure

To assess the capability of the LLMs for medical NER in Urdu, we designed a structured prompt specifying the five target entity categories: *Medication, Symptom, Disease, Treatment, and Body Part*. We refined prompt through preliminary experiments to ensure consistency in model outputs. The final prompt used for evaluation was:

Prompt
"*Extract the following information medication, symptom, disease, treatment, body part from given news summaries. Make a table with headers news, medication, symptom, disease, treatment, body part.*"

3.4 Evaluation Measures

To evaluate the performance of the LLMs on the medical NER task, we used standard entity-level precision, recall, and F1-score metrics. Evaluation was performed by comparing the entities extracted by the LLMs with the manually annotated gold-standard dataset.

We adopted an exact entity-level matching criterion. An extracted entity was considered a ***true positive (TP)*** only if the predicted entity text exactly matched the gold-standard entity span, and the predicted entity was assigned the correct entity category (e.g., Disease, Symptom, Medication, Treatment, or Body Part). If a model extracted an entity that did not appear in the gold-standard annotations, it was counted as a ***false positive (FP).*** If a gold-standard entity was not identified by the model, it was counted as a ***false negative (FN).***

For each entity category, precision, recall, and F1-score were computed. In addition to category level evaluation, we assessed the overall performance using micro-averaging across all entity categories.

4. Results

The annotated dataset contained a total of 2057 Urdu news headlines with brief summaries, categorized into five medical entity types: Disease, Symptom, Treatment, Medication, and Body Part. Table 1 summarizes the annotation statistics, including the total number of annotations and the count of unique entities present in each category.

Table 1: Summary Statistics of Entities in the Urdu News Dataset

Category	**# of Annotations**	**Unique Entities**
Disease	714	333
Symptom	173	129
Medication	102	75
Treatment	106	81
Body Part	745	226

We then analyzed the total number of entities extracted by the LLMs across the five medical categories. Figure 3 presents a comparison between human annotations, ChatGPT-4o, and LLAMA 3.2. The results highlight a notable difference in the count of entities which are extracted by human annotators and LLMs. Across all entity categories, both models generated substantially higher numbers of entity mentions compared to the gold-standard annotations.

For the Disease category, human annotators identified 714 entities, whereas LLAMA 3.2 extracted 1,725 entities and ChatGPT-4o extracted 1,297. In the Symptom category, humans annotated 173 entities, compared to 1,224 extracted by LLAMA 3.2 and 1,350 by ChatGPT-4o. A similar pattern was observed for Medication, where 102 entities were annotated by humans, while LLAMA 3.2 and ChatGPT-4o identified 757 and 631 entities, respectively. In the Treatment category, human annotators labeled

Table 2: Entity-level evaluation results (exact match) for LLAMA 3.2 and ChatGPT-4o on the Urdu medical NER dataset

Entity	Model	TP	FP	FN	Precision	Recall	F1
Disease	LLAMA 3.2	403	1322	311	0.234	0.564	0.33
	ChatGPT-4o	352	945	362	0.271	0.493	**0.35**
Symptom	LLAMA 3.2	73	1151	100	0.06	0.422	**0.105**
	ChatGPT-4o	71	1279	102	0.053	0.41	0.093
Medication	LLAMA 3.2	42	715	60	0.055	0.412	0.098
	ChatGPT-4o	36	595	66	0.057	0.353	0.098
Treatment	LLAMA 3.2	35	1784	71	0.019	0.33	**0.036**
	ChatGPT-4o	10	1660	96	0.006	0.094	0.011
Body Part	LLAMA 3.2	303	1664	442	0.154	0.407	0.223
	ChatGPT-4o	331	1429	414	0.188	0.444	**0.264**
Micro Avg	LLAMA 3.2	856	6636	984	0.114	0.465	0.183
	ChatGPT-4o	**800**	**5908**	**1040**	**0.119**	**0.435**	**0.187**

106 entities, whereas LLAMA 3.2 extracted 1,819 and ChatGPT-4o extracted 1,670 entities. Finally, in the Body Part category, humans annotated 745 entities, compared to 1,967 identified by LLAMA 3.2 and 1,760 by ChatGPT-4o. Overall, both LLMs produced substantially higher entity counts than the human-annotated gold standard.

4.1 Performance Evaluation of LLMs

We compared the performance of the LLMs using entity-level precision, recall, and F1-score against the manually annotated gold-standard labels. Table 2 presents the detailed evaluation results for both LLAMA 3.2 and ChatGPT-4o. Overall, the results indicate low precision but comparatively higher recall across most entity categories, indicating over-generation.

For the *Disease* category, ChatGPT-4o slightly outperformed LLAMA 3.2 in terms of F1-score, achieving 0.35 compared to 0.33. Although LLAMA 3.2 achieved higher recall (0.564 vs 0.493), ChatGPT-4o's higher precision (0.271 vs 0.234) contributed to its superior F1-score. In the *Body Part* category, ChatGPT-4o also outperformed LLAMA 3.2, achieving an F1-score of 0.264 compared to 0.223. ChatGPT-4o demonstrated both higher precision (0.188 vs 0.154) and recall (0.444 vs 0.407).

For *Symptom*, LLAMA 3.2 achieved a slightly higher F1-score (0.105) than ChatGPT-4o (0.093). Recall remained moderate for both models (0.422 for LLAMA 3.2 and 0.41 for ChatGPT-4o), but precision was very low (<0.06), indicating over-generation of entities.

In the *Medication* category, both models achieved similar performance (F1-scores of 0.098). LLAMA 3.2 had slightly higher recall (0.412 vs 0.353), whereas ChatGPT-4o had marginally better precision (0.057 vs 0.055). The *Treatment* category remained the most challenging, with low F1 scores for both models. LLAMA 3.2 achieved 0.036, while ChatGPT-4o only achieved 0.011. The particularly low recall of ChatGPT-4o (0.094) highlights difficulty in identifying treatment-related entities in Urdu medical text.

Micro-averaged performance shows that ChatGPT-4o slightly outperforms LLAMA 3.2 overall. ChatGPT-4o achieved a micro-average F1-score of 0.187 compared to 0.183 for LLAMA 3.2. Despite both models exhibiting low precision (0.119 vs 0.114), their moderate recall (0.435 for ChatGPT-4o and 0.465 for LLAMA 3.2) highlights

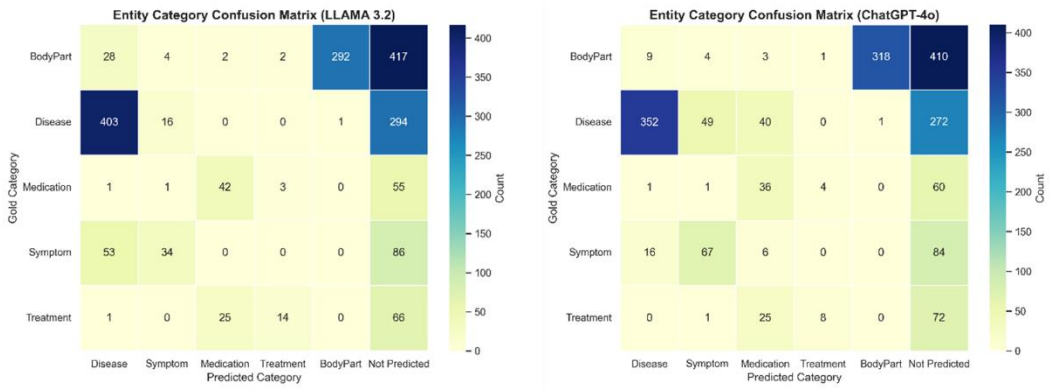


Figure 4: Confusion matrices comparing predicted versus gold-standard entity categories for LLAMA 3.2 and ChatGPT-4o on the Urdu medical NER dataset

the general tendency of LLMs to over-predict entities across all categories.

4.2 Error Analysis

We conducted an error analysis at the entity level to better understand the performance patterns of both LLMs. False positives (FPs) and false negatives (FNs) were examined. The analysis showed that following were the common types of mistakes made by LLMs in the Urdu medical domain:

4.2.1 Over generation of general terms (False Positives)

Both LLMs tend to over-generate general or common terms that are not specific entities in the gold standard. LLAMA 3.2 frequently predicted terms like "عام صحت" *(general health)* and " مختلف بیماریاں" (different diseases) as 'Disease', " علاج کرنا" (*do treatment*) as 'Treatment'. ChatGPT-4o over-predicted similar generic terms such as "مختلف امراض" *(various diseases*) as 'Disease' and "عمومی صحت" (*General Health*) as 'Medication'. These patterns highlight a tendency of both models to capture broadly related concepts rather than the precise entities defined in the annotated dataset.

4.2.2 Missed Critical Entities (False Negatives)

Both models sometimes failed to identify clinically important entities. LLAMA 3.2 missed key Disease entities like "کینسر" (*cancer*) and "موٹاپے" (*obesity*). ChatGPT-4o also missed similar entities in Disease, including "موٹاپے" (*obesity)*, "کینسر" (*cancer*) and "ذیابیطس" (*diabetes*). For Symptom, Medication, Treatment, and Body Part categories, both models missed essential entities such as "سردرد" (*headache, Symptom)*, *"* انسولین *"* *(insulin, Medication)*, "ہڈیوں" *(bones, Body Part).*

4.2.3 Morphological and Contextual Variation Challenges

Both LLMs struggled with handling morphological variations and context-specific forms in Urdu medical text. Several examples illustrate this limitation:

Pluralization issues: In a headline containing "دماغوں" (brains, plural), both LLAMA 3.2 and ChatGPT-4o extracted only "دماغ" (brain, singular), resulting in a mismatch with the gold-standard entity. Similarly, the plural "موٹاپے" (obesity, plural form) was normalized to "موٹاپا" (singular) by the models.

Compound and context-specific terms: In the phrase "دماغی خلیات" (brain cells), only "دماغ" was identified by the models, while the full entity was not captured.

4.2.4 Category Confusion Patterns

The confusion matrices shown in Figure 4 highlight systematic cross-category errors:

4.2.4.1 Body Part vs Disease/Symptom:
LLAMA 3.2 misclassified 28 Body Part entities as Disease, while ChatGPT-4o misclassified only 9.

4.2.4.2 Symptom vs Disease
Both models confused Symptom and Disease entities, with 53 (LLAMA 3.2) and 16 (ChatGPT-4o) Symptom entities predicted as Disease.

4.2.4.3 Treatment vs Medication
Treatment entities were often misclassified as Medication (14 instances for LLAMA 3.2, 25 for ChatGPT-4o).

These patterns indicate that both models not only over-generate general terms but also struggle to correctly assign entities to the appropriate category, particularly when categories share semantic similarity (e.g., Disease vs Symptom).

5. Discussion

This study evaluated the performance of two state-of-the-art LLMs, ChatGPT-4o and LLAMA 3.2, for medical NER in Urdu medical text. The findings highlight both the potential and the limitations of LLMs in handling low-resource languages like Urdu, particularly in the medical domain. While ChatGPT-4o achieved slightly better performance in terms of micro-averaged precision, recall, and F1-score (Micro F1: 0.187) compared to LLAMA 3.2 (Micro F1: 0.183), the

overall moderate scores for both models underscore the need for more specialized training and domain-specific optimization.

The annotation statistics provided valuable insights into model behavior. Both LLMs extracted substantially more entities than human annotators, often resulting in over-generation. This trend was particularly notable in the Body Part category, where ChatGPT-4o extracted 2,242 entities and LLAMA 3.2 extracted 2,105, compared to 745 annotated by humans. Such over-generation reflects the models' tendency to overgeneralize linguistic patterns learned during pretraining, leading to false positives, especially for generic or common terms. For instance, LLAMA 3.2 repeatedly predicted the term "جسم" (body) as a Body Part entity, occurring 796 times, while ChatGPT-4o predicted it 411 times. This illustrates the tendency of both models to over-generate common or generic terms, contributing significantly to false positives.

Beyond over-generation, both models exhibited challenges in handling morphological and variations. Cases such as "دماغوں" (brains) versus "دماغ" (brain), "دماغی خلیات" (brain cells) versus "دماغ", and "موٹاپے" versus "موٹاپا" demonstrate that LLMs often failed to fully capture plural forms, and compounds. This also highlights a language-specific challenge inherent to Urdu, which is characterized by rich morphology and complex word formations.

Performance varied across entity categories. ChatGPT-4o achieved higher F1-scores in Disease (0.35), Body Part (0.264), and slightly in Medication (0.098), while LLAMA 3.2 outperformed in Symptom (0.105). Both models struggled with Treatment, achieving F1-scores of 0.036 (LLAMA 3.2) and 0.011 (ChatGPT-4o). This could be attributed to the inherent ambiguity of treatment-related terminology in Urdu, which often involves multi-word expressions such as " گامانائف آئی کون مشین" (*Gama Knife Icon Machine*). These challenges are further exacerbated by the limited availability of pretraining data in the healthcare domain for Urdu, which restricts the models' ability to accurately disambiguate nuanced medical terms. A similar study (Naguib et al., 2024) also highlights these challenges where low-resource languages and domain-specific tasks showed suboptimal performance due to limited linguistic and domain coverage in pretraining corpora.

Overall, the results underscore the inherent challenges of medical NER in low-resource languages. Unlike high-resource languages such as English, which benefit from extensive annotated datasets and domain-specific corpora, Urdu suffers from limited labeled data, diverse linguistic structures, and terminological variations. These limitations contributed to low precision and moderate recall, even when models captured many entities. Nonetheless, the study demonstrates the feasibility of applying LLMs to medical NER in Urdu. ChatGPT-4o showed a slight edge in overall micro-averaged performance.

The performance of the LLMs could be potentially improved using fine tuning and few-shot learning techniques (Moscato et al., 2023). These techniques have shown promising improvements in the medical NER task in other languages (Rohanian et al., 2024; Zhu et al., 2024). Future work should focus on morphology-aware preprocessing to handle plurals and compound entities, domain-specific fine-tuning to reduce over-generation, and refined prompt engineering. These approaches, combined with few-shot and transfer learning, have potential for more accurate and scalable medical NER in Urdu and other low-resource languages.

6. Conclusion

This study evaluated the performance of two state-of-the-art LLMs, ChatGPT-4o and LLAMA 3.2, on medical NER in Urdu language. ChatGPT-4o generally outperformed LLAMA 3.2 across most metrics, highlighting its relative strength in extracting medical entities from low-resource language text. Despite this, both models demonstrated only moderate performance, reflecting challenges such as over-generation, difficulty with medical terminologies, and limited pretraining data in the Urdu medical domain. These findings underscore the need for targeted fine-tuning, morphology-aware preprocessing, and the development of larger, high-quality annotated datasets to improve LLM effectiveness. Overall, this study provides a foundational benchmark for applying NLP techniques to medical text in low-resource languages and informs future research aimed at enhancing LLM-based NER in similar contexts.

7. Acknowledgments

None.

8. Data Availability Statement

The data that support the findings of this study are available from the corresponding author (zarmeen.nasim@aku.edu) upon reasonable request.

9. Bibliographical References

Ahmed, A., Huang, D., & Arafat, S. Y. (2024). Enriching Urdu NER with BERT Embedding, Data Augmentation, and Hybrid Encoder-CNN Architecture. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *23*(4). https://doi.org/10.1145/3648362

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., &

McDermott, M. B. A. (2019). *Publicly Available Clinical BERT Embeddings*. https://arxiv.org/abs/1904.03323v3

Anam, R., Anwar, M. W., Jamal, M. H., Bajwa, U. I., De la Torre Diez, I., Alvarado, E. S., Flores, E. S., & Ashraf, I. (2024). A deep learning approach for Named Entity Recognition in Urdu language. *PLOS ONE*, *19*(3), e0300725. https://doi.org/10.1371/JOURNAL.PONE.0300725

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. El, Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., … Wu, Y. (2023). *PaLM 2 Technical Report*. https://arxiv.org/abs/2305.10403v3

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Coleman, J., Krishnamachari, B., Iskarous, K., & Rosales, R. (2024). *LLM-Assisted Rule Based Machine Translation for Low/No-Resource Languages*. 67–87. https://doi.org/10.18653/v1/2024.americasnlp-1.9

Deng, X., Bashlovkina, V., Han, F., Baumgartner, S., & Bendersky, M. (2023). LLMs to the Moon? Reddit Market Sentiment Analysis with Large Language Models. *ACM Web Conference 2023 - Companion of the World Wide Web Conference, WWW 2023*, 1014–1019. https://doi.org/10.1145/3543873.3587605

Elshin, D., Karpachev, N., Gruzdev, B., Golovanov, I., Ivanov, G., Antonov, A., Skachkov, N., Latypova, E., Layner, V., Enikeeva, E., Popov, D., Chekashev, A., Negodin, V., Frantsuzova, V., Chernyshev, A., & Denisov, K. (2024). *From General LLM to Translation: How We Dramatically Improve Translation Quality Using Human Evaluation Data for LLM Finetuning* (pp. 247–252). https://aclanthology.org/2024.wmt-1.17

Iwakami, Y., Takuma, H., & Iwashita, M. (2024). Complementary Method for NER by Extracting Proper Nouns from Images when Utilizing LLM. *2024 IEEE/ACIS 9th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, 233–238. https://doi.org/10.1109/BCD61269.2024.10743114

Jahan, I., Laskar, M. T. R., Peng, C., & Huang, J. X. (2024). A comprehensive evaluation of large Language models on benchmark biomedical text processing tasks. *Computers in Biology and Medicine*, *171*. https://doi.org/10.1016/j.compbiomed.2024.108189

Kumichev, G., Blinov, P., Kuzkina, Y., Goncharov, V., Zubkova, G., Zenovkin, N., Goncharov, A., & Savchenko, A. (2024). MedSyn: LLM-Based Synthetic Medical Text Generation Framework. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *14950 LNAI*, 215–230. https://doi.org/10.1007/978-3-031-70381-2_14

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 260–270. https://doi.org/10.18653/V1/N16-1030

Leaman, R., Khare, R., & Lu, Z. (2015). Challenges in clinical natural language processing for automated disorder normalization. *Journal of Biomedical Informatics*, *57*, 28–37. https://doi.org/10.1016/J.JBI.2015.07.010

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020a). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240. https://doi.org/10.1093/BIOINFORMATICS/BTZ682

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020b). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240. https://doi.org/10.1093/BIOINFORMATICS/BTZ682

Moscato, V., Postiglione, M., & Sperlí, G. (2023). Few-shot Named Entity Recognition: Definition, Taxonomy and Research Directions. *ACM Transactions on Intelligent Systems and Technology*, *14*(5). https://doi.org/10.1145/3609483/ASSET/CD30433D-FB6D-4CEE-BDA4-86AC6442DA2D/ASSETS/GRAPHIC/TIST-2022-10-0375-F14.JPG

Naguib, M., Tannier, X., & Névéol, A. (2024). Few-shot clinical entity recognition in English, French and Spanish: masked language models outperform generative model prompting. *Proceedings of the 2nd. Clinical Natural Language Processing*

Workshop, 6829–6852.

Nasim, Z., Abidi, S., & Haider, S. (2020). Modeling POS Tagging for the Urdu Language. *2020 International Conference on Emerging Trends in Smart Technologies, ICETST 2020*. https://doi.org/10.1109/ICETST49965.2020.9080721

Nasim, Z., & Haider, S. (2022). Automatic Labeling of Clusters for a Low-Resource Urdu Language. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *21*(5). https://doi.org/10.1145/3511097/ASSET/75F5299D-108C-4D86-919D-9CD617162B10/ASSETS/GRAPHIC/TALLIP-20-0576-INLINE10.GIF

Naz, S., Umar, A. I., Shirazi, S. H., Khan, S. A., Ahmed, I., & Khan, A. A. (2014). Challenges of Urdu named entity recognition: A scarce resourced language. *Research Journal of Applied Sciences, Engineering and Technology*, *8*(10), 1272–1278. https://doi.org/10.19026/RJASET.8.1095

Rohanian, O., Nouriborji, M., Kouchaki, S., Nooralahzadeh, F., Clifton, L., & Clifton, D. A. (2024). Exploring the effectiveness of instruction tuning in biomedical language processing. *Artificial Intelligence in Medicine*, *158*, 103007. https://doi.org/10.1016/J.ARTMED.2024.103007

Wang, Z., Zhou, Q., Junfeng, Z., Wang, Y., Ding, H., & Song, J. (2023). A Knowledge-Enhanced Medical Named Entity Recognition Method that Integrates Pre-Trained Language Models. *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, 296–301. https://doi.org/10.1109/MEDAI59581.2023.00046

Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q. L., & Tang, Y. (2023). A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development. *IEEE/CAA Journal of Automatica Sinica*, *10*(5), 1122–1136. https://doi.org/10.1109/JAS.2023.123618

Zhang, M., & Li, J. (2021). A commentary of GPT-3 in MIT Technology Review 2021. *Fundamental Research*, *1*(6), 831–833. https://doi.org/10.1016/J.FMRE.2021.11.011

Zhao, Y., Wang, C., & Fu, G. (2012). A CRF Sequence Labeling Approach to Chinese Punctuation Prediction. *Pacific Asia Conference on Language, Information and Computation*.

Zhu, Z., Zhao, Q., Li, J., Ge, Y., Ding, X., Gu, T., Zou, J., Lv, S., Wang, S., & Yang, J. J. (2024). Comparative Analysis of Large Language Models in Chinese Medical Named Entity Recognition. *Bioengineering 2024, Vol. 11, Page 982*, *11*(10), 982. https://doi.org/10.3390/BIOENGINEERING11100982

DR-RAG: Addressing Retrieval Misalignment in Low-Resource Urdu Question Answering

Saad Ahmad, Muhammad Hammad, Muhammad Zeeshan, Faizad Ullah, Asim Karim
Department of Computer Science
Lahore University of Management Sciences (LUMS), Lahore, Pakistan
{23030014, 23030013, 23030011, faizadullah, akarim}@lums.edu.pk

Abstract

Retrieval-Augmented Generation performs well on English QA benchmarks, but degrades considerably in morphologically rich, low-resource languages. Urdu presents a particularly challenging case: heavy inflectional morphology, Nastaliq script inconsistencies, and limited training data produce a systematic mismatch between query representations and indexed document content that standard retrieval architectures cannot bridge. We propose **DR-RAG** (Dual-Representation Retrieval-Augmented Generation), which addresses this through dual indexing. Each document is represented as overlapping text chunks and as automatically generated question-answer pairs. Queries are first matched against the QA index, which aligns more reliably with natural query phrasing than declarative document chunks. When retrieval confidence falls below $\tau = 0.80$, the system falls back to chunk-based retrieval, maintaining coverage without sacrificing precision. Evaluated on Urdu UQA and English SQuAD 2.0, DR-RAG improves Urdu METEOR by $38\times$, ROUGE-1 by 140%, and reduces generation latency by 43%. LLM-as-judge scores show higher faithfulness (3.03 vs 1.93) and overall quality (2.99 vs 2.21) over MultiVector. English performance remains competitive throughout. These results indicate that representation-level alignment between queries and indexed content, rather than increased model complexity, is the critical factor for reliable retrieval in underserved South Asian languages. Code: https://github.com/saadahmad17/DR_RAG.

Keywords: Low-resource NLP, Urdu question answering, Retrieval-augmented generation, South Asian languages, Morphological complexity, Dual-representation retrieval

1. Introduction

Retrieval-Augmented Generation has become a standard approach for grounding language model outputs in external knowledge, yet its development has remained disproportionately centred on English. The assumptions underlying most RAG pipelines, well-calibrated dense embeddings, consistent tokenization, and morphologically stable text, hold reasonably well for English but break down in languages where these conditions are not met. Urdu is precisely such a language, and building a functional Urdu QA system exposes these assumptions in concrete terms.

Large Language Models have achieved strong results across summarization, question answering, and dialogue (Brown et al., 2020), but retain well-documented limitations: knowledge is fixed at training time, factual hallucination remains a persistent problem, and domain-specific or time-sensitive information is inaccessible without external retrieval (Petroni et al., 2019; Shuster et al., 2021). RAG addresses part of this by grounding generation in retrieved evidence (Lewis et al., 2020; Izacard and Grave, 2020). In low-resource settings, however, the retrieval component itself becomes the bottleneck. For Urdu specifically, tokenization errors and weaker semantic alignment substantially depress retrieval precision (Khanuja et al., 2021). A query using one inflected form of a word will not reliably match a document passage using a different inflected form of the same word, even when both refer to identical content.

Multi-vector retrieval, as implemented in LangChain's MultiVector retriever, attempts to address this through richer document representation. Our experiments show that while this approach improves retrieval in English, it degrades to near-zero precision on Urdu. The morphological and orthographic gap between query and document is sufficiently large that English-tuned retrieval strategies cannot bridge it reliably.

We introduce **DR-RAG** (Dual-Representation Retrieval-Augmented Generation) as a linguistically motivated response to this failure mode. The framework indexes each document in two complementary forms: overlapping text chunks and automatically generated question-answer (QA) pairs. The QA index functions as a query-language resource, storing document content in a syntactic form that more closely mirrors user query structure, thereby reducing the representational distance that causes retrieval to fail in Urdu. A confidence-aware fallback mechanism routes queries to the QA index first, switching to chunk retrieval only when the top similarity score falls below a set threshold. Language-specific choices in chunk sizing and embedding

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 49–58
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

models additionally address the tokenization noise characteristic of low-resource Urdu text.

Evaluated on Urdu UQA (Arif et al., 2024) and English SQuAD 2.0 (Rajpurkar et al., 2018), DR-RAG achieves a $38\times$ METEOR improvement, 140% ROUGE-1 gain, and 43% latency reduction on Urdu, while remaining competitive on English. Our contributions are as follows:

1. We provide a systematic analysis of retrieval misalignment in Urdu QA, attributing it to morphological inflection, orthographic variability, and inadequate low-resource embedding coverage.
2. We propose DR-RAG, a dual-representation retrieval framework with a confidence-aware fallback policy specifically designed to address retrieval misalignment in morphologically rich, low-resource languages.
3. We demonstrate through evaluation on UQA and SQuAD 2.0 that linguistically motivated design choices, rather than increased architectural complexity, are the primary driver of reliable performance gains in South Asian language question answering.

2. Related Work

2.1. Retrieval-Augmented Generation

RAG was originally proposed to mitigate factual hallucination in large language models by conditioning generation on retrieved evidence (Lewis et al., 2020). Subsequent work has extended this foundation in several directions. Hybrid retrieval combines sparse BM25 with dense vector search to improve robustness across domains (Mandikal and Mooney, 2024), while modular architectures incorporate query rewriting and reranking as task-dependent components (Yu et al., 2023). Adaptive retrieval methods introduce selectivity: DRAGIN conditions retrieval on model uncertainty (Su et al., 2024), reinforcement learning policies optimise the accuracy-efficiency trade-off (Shuster et al., 2024), and MemoRAG caches prior retrievals to support multi-step reasoning (Qian et al., 2025). For complex multi-document reasoning, layered retrieval architectures decompose queries into tractable subproblems (Li et al., 2025; Zhang et al., 2025), while passage integration models such as FiDO (De Jong et al., 2023) and ColBERT-RAG (Santhanam et al., 2022) improve cross-passage comparison during decoding. Structured evaluation frameworks like RAGAS (Rajagopal et al., 2023) now enable systematic assessment of faithfulness and contextual relevance.

Despite this considerable progress, virtually all RAG research has been conducted on English benchmarks. The extent to which these advances transfer to morphologically rich, low-resource languages, where embedding quality is lower and linguistic preprocessing tools are limited, remains largely uninvestigated. This work addresses that gap directly, using Urdu as the primary test case.

2.2. Urdu NLP and RAG

Urdu NLP has historically been constrained by the absence of large-scale annotated resources. The release of the UQA dataset (Arif et al., 2024) represented a significant step forward, providing a benchmark for both extractive and generative question answering in Urdu. Tahir et al. (Tahir et al., 2024) subsequently evaluated a range of multilingual models on UQA, documenting consistent and substantial performance gaps relative to English, gaps that motivate the present work directly.

Recent model development has made further progress possible. UrduLLaMA 1.0 (Fiaz et al., 2025) demonstrates that instruction tuning on native Urdu corpora yields meaningful improvements over general multilingual models such as mBERT and XLM-R. Alif (AI, 2025) extends this further through large-scale Urdu-centric pretraining. However, even these dedicated models exhibit weaknesses in factual consistency and multi-hop reasoning, indicating that stronger generation capacity alone cannot compensate for retrieval-level failures. Linguistic preprocessing resources including UrduHack (Contributors, 2021) and LughaatNLP (Farooq and Khan, 2023) provide morphological normalisation support that underpins this work. Recent findings confirm that adaptation gaps for South Asian languages persist even with modern instruction-tuned LLMs (Khade et al., 2025).

To our knowledge, no prior work has examined the specific mechanisms by which standard RAG retrieval fails in Urdu, nor proposed an architecture explicitly designed to address those mechanisms.

2.3. Urdu Linguistic Challenges for RAG

Three structural properties of Urdu create specific and compounding difficulties for retrieval-based systems.

Morphological Complexity. Urdu is highly inflectional, marking gender, number, and case through suffixation. A single lexical root can yield multiple orthographically distinct surface forms across grammatical contexts: the verb meaning “to write”, for instance, surfaces as *likha*, *likhi*, *likhta*, or *likhti* depending on subject agreement. Both BM25 and dense embedding models operate on surface forms, meaning that a query using one inflected variant will not reliably retrieve a document passage using a different variant of the same root. In Urdu,

this is not an edge case but a pervasive retrieval failure mode.

Script and Orthographic Variation. Urdu is written in Nastaliq, a connected cursive script in which word boundaries are often ambiguous and optional spacing alters tokenization in unpredictable ways. Diacritical marks, zabar, zer, and pesh, which disambiguate phonemic and occasionally semantic distinctions, are routinely omitted in digital text.

Multilingual tokenizers, optimised predominantly for high-resource languages, handle this variation inconsistently, producing embeddings that fail to generalise across orthographic forms of the same token (Khanuja et al., 2021). Specifically, this ambiguity manifests at the tokenization level in digital text processing: tokenizers not designed with Nastaliq's cursive connectivity in mind frequently mis-segment word boundaries, producing inconsistent token sequences for the same lexical item across different orthographic renderings. We characterise this as *embedding brittleness*: the tendency of Urdu token representations to fragment across surface variants rather than converging on stable semantic representations.

Low-Resource Embedding Weakness. Urdu is substantially underrepresented in the pretraining corpora of most multilingual embedding models. Dense retrieval requires a well-calibrated semantic space, and that calibration is data-dependent. BM25 faces a parallel problem: vocabulary coverage is limited and reliable morphological normalisation tools are scarce, leaving both retrieval paradigms less effective than they would be for a high-resource language (Tahir et al., 2024).

Implications for Retrieval Design. These properties combine to produce a query-document mismatch that operates at the level of surface form rather than semantic content. A user query and the relevant document passage may refer to identical information while appearing sufficiently different to current retrieval systems to go unmatched. DR-RAG addresses this by indexing content in question form, which preserves the syntactic structure of natural queries and substantially reduces the representational distance that causes retrieval to fail in Urdu.

3. Methodology

DR-RAG (Dual-Representation Retrieval-Augmented Generation) indexes each document twice: as overlapping text chunks and as automatically generated question-answer pairs. At query time, the QA index is searched first, since questions align more naturally with other questions than with raw document text. If no confident match is found, the system falls back to chunk-based retrieval. Figure 1 shows the complete three-phase pipeline.

3.1. Document Processing and Indexing

Each document is split into overlapping chunks of 200-500 tokens using a recursive splitter that preserves continuity across boundaries. Overlapping chunks matter especially for Urdu, where morphological context often spans clause boundaries and losing that carryover weakens semantic understanding.

From each chunk, multiple QA pairs are generated to create a semantically dense secondary index. For English, QA pairs are produced using LLaMA 2 7B (Touvron et al., 2023). For Urdu, GPT-4o is used instead. Pilot tests showed that Urdu-specific models such as Alif (AI, 2025), GLM-4, and AYA produced inconsistent outputs with weaker factual alignment, consistent with findings in (Arif et al., 2024) that general-purpose models outperform Urdu-specific ones on instruction-following tasks.

Generated QA pairs were subsequently reviewed by professional Urdu speakers to verify factual accuracy and filter hallucinated or culturally inconsistent content prior to indexing, ensuring the quality of the dual-representation index.

Embedding and Vectorization. Chunks and QA pairs are embedded using language-appropriate encoders. For English, `all-MiniLM-L6-v2` (Wang et al., 2020) provides lightweight but reliable semantic representations. For Urdu, `intfloat/multilingual-e5-large` (Wang et al., 2024b) performs better given its stronger multilingual alignment for low-resource settings. All vectors are normalised before indexing.

Vector Indexing. Embeddings are stored in `Qdrant` (Team, 2023), a vector database built for efficient approximate nearest neighbour search. Separate indices for QA pairs and chunks allow the system to switch between representations at query time.

3.2. Retrieval and Answer Generation

Primary Retrieval. Incoming queries are embedded and matched against the QA index using cosine similarity. Question-form representations produce better matches for user queries than declarative document chunks do, particularly in Urdu where the morphological gap between query and passage can be wide.

Confidence-Aware Retrieval Policy. When the top QA match falls below $\tau = 0.80$, the system switches to chunk-based retrieval. This threshold

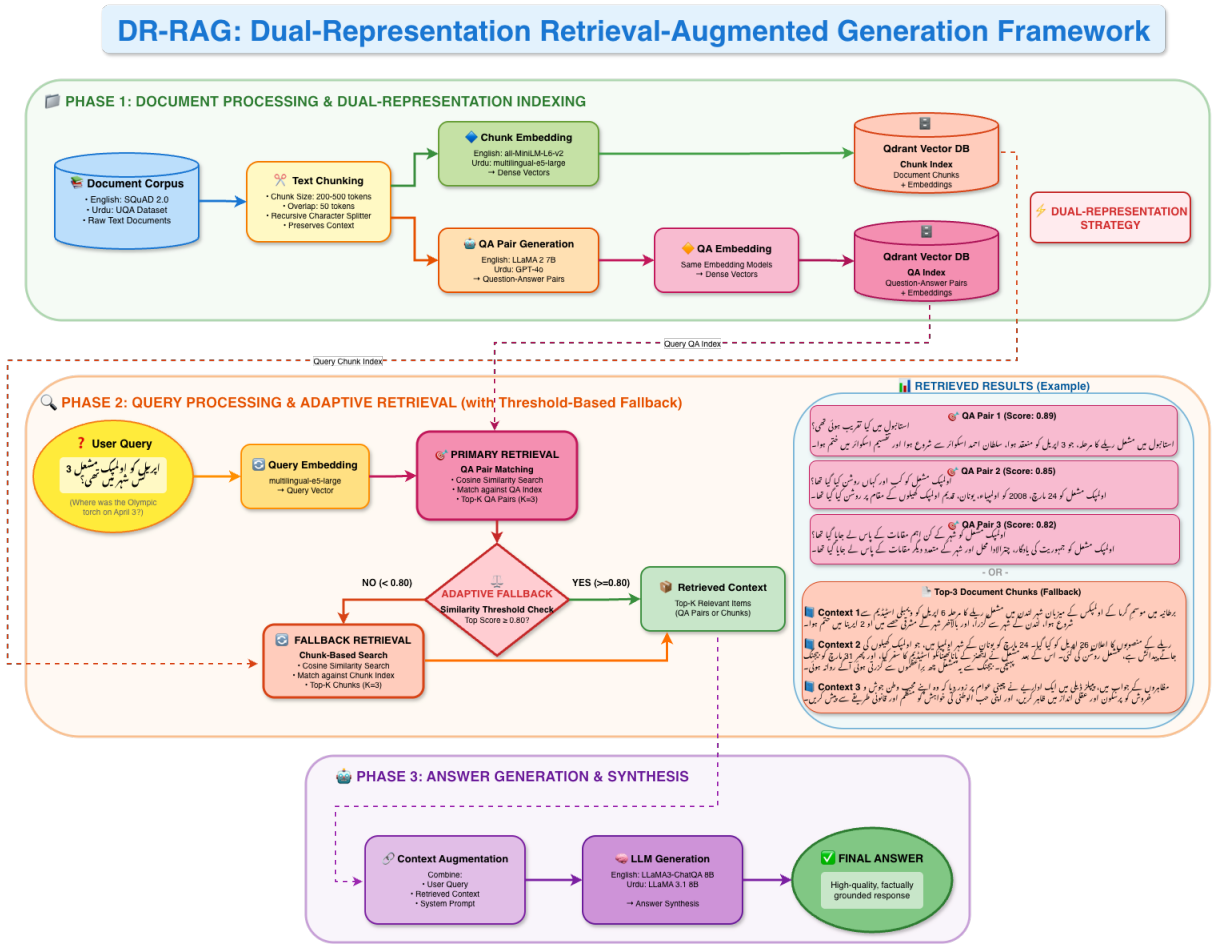


Figure 1: DR-RAG system architecture showing the dual-representation retrieval pipeline. **Phase 1: Document Processing**—Documents are split into overlapping chunks while an LLM generates question–answer pairs from the same content. Both are embedded (`all-MiniLM-L6-v2` for English, `multilingual-e5-large` for Urdu) and indexed separately. **Phase 2: Confidence-Aware Retrieval**—Queries first search the QA index. If the best match scores above $\tau = 0.80$, that context is used; otherwise the system falls back to document chunks. **Phase 3: Answer Generation**—Retrieved context feeds into LLaMA3-ChatQA 8B (English) or LLaMA 3.1 8B (Urdu) to produce the final answer. This dual-index design with confidence-aware fallback improves both precision and coverage, especially for morphologically complex low-resource languages such as Urdu.

was selected empirically based on observed confidence score distributions across the evaluation set, where scores above 0.80 consistently corresponded to semantically relevant matches. This keeps the system functional even when QA pairs are sparse or poorly matched, something that static multi-vector systems like LangChain's default implementation (Chase and Contributors, 2023) cannot do.

Answer Synthesis. Retrieved context is passed to a generative model for final answer production. LLaMA3-ChatQA 8B handles English queries and LLaMA 3.1 8B handles Urdu, as it better accommodates Urdu morphology.

3.3. Comparative Baselines

Traditional RAG. A standard chunk-only retriever that finds the top-3 most similar chunks via cosine similarity and passes them to the generator. This represents the common deployment baseline for low-resource QA.

LangChain MultiVector. Uses hierarchical parent-child indexing, where document chunks are parents and derived representations such as summaries or QA pairs are children. This gives richer semantic coverage than flat chunk retrieval. However, the retrieval policy is static with no confidence-based switching, which our experiments show causes it to collapse on Urdu (Chase and Contributors, 2023).

3.4. Evaluation Framework

Performance is measured across four dimensions: lexical accuracy, semantic alignment, retrieval effectiveness, and computational efficiency.

Automatic Metrics. BLEU, ROUGE, METEOR, and SacreBLEU measure lexical overlap with reference answers. We acknowledge these metrics have known limitations for morphologically rich languages and complement them with BERTScore for semantic alignment and two factuality indicators: Faithfulness (consistency with retrieved evidence) and Answer Relevancy (how directly the response addresses the query). Retrieval quality is assessed via Context Precision, Context Recall, and Entity Recall.

LLM-as-Judge. GPT-3.5 Turbo rates each response on a 1-5 scale across Faithfulness, Correctness, Relevance, Conciseness, and Overall Quality. This gives a human-aligned quality signal that automatic metrics alone cannot capture.

Latency. Retrieval and generation times are recorded separately. For low-resource deployment contexts, where hardware is often constrained, latency is not just a performance metric but a practical deployment consideration.

3.5. Evaluation Setup

Hardware and Software. Experiments ran on a local workstation with an NVIDIA GTX 1080 (8 GB VRAM) and 24 GB RAM, using `PyTorch`, `Hugging Face Transformers`, `LangChain`, and `Qdrant`. Model quantization was applied to reduce memory load and inference latency.

Datasets. We used SQuAD 2.0 (Rajpurkar et al., 2018) for English and UQA (Arif et al., 2024) for Urdu. From a corpus of 440 topic-aligned PDF documents, five were indexed for English and two for Urdu, reflecting the genuine data scarcity in the Urdu setting. Models were tested on 1,200 English and 900 Urdu queries.

Compared Systems.

- **Traditional RAG:** Top-3 chunk retrieval via cosine similarity.
- **LangChain MultiVector:** Parent-child indexing without adaptive fallback.
- **DR-RAG:** QA-first retrieval with confidence-aware fallback at $\tau = 0.80$.

3.6. Retrieval Quality

LangChain MultiVector is competitive in English but falls apart on Urdu, recording precision of 0.001 and recall of 0.004. This is not a minor gap but a near-total failure, confirming that English-optimised retrieval does not transfer to morphologically complex languages (Tahir et al., 2024).

Metric	Trad. RAG	MultiVector	DR-RAG
Context Precision			
English	0.0402	**0.0710**	0.0495
Urdu	0.0311	0.0010	**0.0672**
Context Recall			
English	**0.4555**	0.3340	0.3697
Urdu	0.2880	0.0040	**0.4023**
Entity Recall			
English	**0.3386**	0.2750	0.2968
Urdu	0.2042	0.0450	**0.3331**

Table 1: Retrieval metrics for English and Urdu. DR-RAG leads on Urdu across all three metrics. LangChain MultiVector collapses to near-zero on Urdu, confirming it is not suited for low-resource settings.

Metric	Trad. RAG	MultiVector	DR-RAG
BERTScore F1	0.7955	0.8057	**0.8134**
BLEU	1.5424	4.7063	**4.8860**
ROUGE-1	0.0660	0.1569	**0.1584**
ROUGE-2	0.0211	**0.0890**	0.0817
ROUGE-L	0.0652	0.1549	**0.1577**
Faithfulness	0.8150	0.8480	**0.8760**
Answer Relevancy	**0.8650**	0.8960	0.8470
METEOR	0.0056	**0.2555**	0.2134
SacreBLEU	2.5278	5.8257	**7.2140**

Table 2: Urdu QA results. DR-RAG leads on most metrics. MultiVector scores higher on METEOR (0.2555 vs 0.2134) but DR-RAG leads on BLEU, SacreBLEU, BERTScore, ROUGE-1, ROUGE-L, and Faithfulness.

DR-RAG reaches Urdu precision of 0.067 and recall of 0.402, a more than sixtyfold precision improvement over MultiVector. Storing content as QA pairs reduces the inflectional variability that breaks chunk-only retrieval, giving the system a stable semantic target even when embeddings are trained on limited data.

3.7. Generation Quality

3.7.1. Urdu Generation Performance

Compared to Traditional RAG, DR-RAG improves ROUGE-1 by 140%, METEOR by nearly $38\times$, and BERTScore by around 2%. The METEOR gain is the most telling: that metric rewards systems that handle morphological variation and synonymy, which are the two core difficulties in Urdu. The QA-pair index appears to normalise inflectional diversity

Metric	Trad. RAG	MultiVector	DR-RAG
BERTScore F1	0.8632	0.1903	**0.8681**
BLEU	0.0694	**0.1134**	0.0730
ROUGE-1	0.2807	**0.3262**	0.3003
ROUGE-2	0.1759	**0.2115**	0.1857
ROUGE-L	0.2771	**0.3249**	0.2960
METEOR	0.4210	**0.4747**	0.4350
Faithfulness	**0.5377**	0.5570	0.5292
Answer Relevancy	**0.6555**	0.7150	0.6033
SacreBLEU	0.0950	**0.1504**	0.0980

Table 3: English QA results. MultiVector leads on lexical overlap. DR-RAG achieves the highest BERTScore, confirming no regression on high-resource data.

System	Retrieval (s)	Generation (s)	Total (s)
English			
Traditional RAG	2.851	4.933	7.784
LangChain MultiVector	**0.074**	1.038	1.112
DR-RAG	2.837	**4.799**	**7.637**
Urdu			
Traditional RAG	**6.793**	14.461	21.255
LangChain MultiVector	0.066	1.465	1.532
DR-RAG	6.846	**8.303**	**15.149**

Table 4: Latency results. DR-RAG cuts Urdu generation time by 43% over Traditional RAG while also improving output quality.

in a way that raw chunk retrieval simply cannot.

Faithfulness rises from 0.815 to 0.876, meaning DR-RAG's answers stay closer to the retrieved evidence. MultiVector scores higher on METEOR (0.2555 vs 0.2134), but DR-RAG leads on the majority of metrics including the factual grounding ones that matter most for QA reliability.

3.7.2. English Generation Performance

On English SQuAD, MultiVector leads on lexical metrics (ROUGE-1: 0.326, BLEU: 0.113), which reflects its English-tuned retrieval favouring surface token overlap. DR-RAG achieves the best BERTScore (0.868), showing stronger semantic alignment even when exact wording differs. Faithfulness scores are close across all systems (0.54-0.56), which makes sense: English has mature embeddings and tokenization tools that already handle factual grounding well. These results confirm DR-RAG is competitive on high-resource data while delivering its main gains in Urdu.

3.8. Latency and Efficiency

MultiVector is the fastest overall, but this appears to reflect implementation optimisation rather than any architectural advantage. DR-RAG's main efficiency gain is in Urdu generation: 8.3s versus 14.5s for Traditional RAG, a 43% reduction. Better retrieval produces tighter context, and the generator processes fewer irrelevant tokens as a result. For real-world South Asian language deployments, where hardware is often constrained, this kind of efficiency gain is as important as accuracy.

System	Faith.	Correct.	Relev.	Concise.	Overall
Urdu					
Traditional RAG	2.208	**2.617**	2.925	2.508	2.565
LangChain MultiVector	1.925	2.158	2.500	2.267	2.212
DR-RAG	**3.033**	2.608	**3.400**	**2.917**	**2.990**
English					
Traditional RAG	3.033	2.267	2.525	2.808	2.658
DR-RAG	**3.067**	2.242	2.867	2.992	2.792
LangChain MultiVector	3.167	**2.717**	**2.917**	**3.267**	**3.017**

Table 5: LLM-as-Judge scores (1-5). DR-RAG leads on all Urdu dimensions and maintains the highest Faithfulness in English.

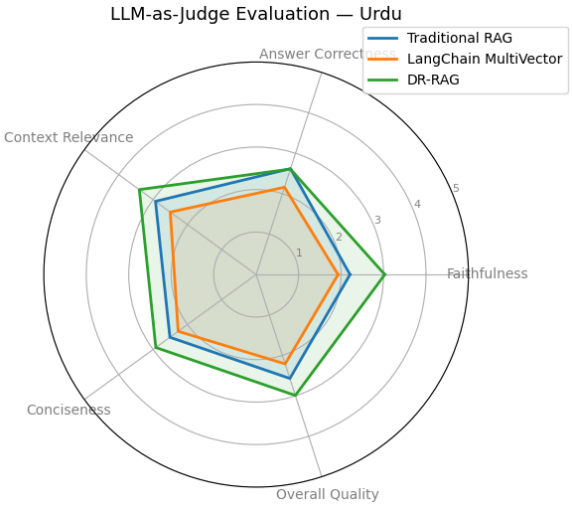


Figure 2: LLM-as-Judge radar chart for Urdu QA. DR-RAG leads across all qualitative dimensions.

3.9. LLM-as-Judge Evaluation

GPT-3.5 Turbo rated each response on Faithfulness, Correctness, Relevance, Conciseness, and Overall Quality (1-5 scale). For Urdu, DR-RAG leads on every dimension. The 57% Faithfulness advantage over MultiVector (3.03 vs 1.93) is the most important result here: it shows the system's answers stay grounded in retrieved evidence rather than drifting into hallucination.

In English, MultiVector scores higher on Overall Quality (3.02 vs 2.79) and Conciseness, consistent with its English-tuned pipeline. DR-RAG holds the highest Faithfulness (3.07). For applications where evidence traceability matters more than brevity, that is the right trade-off to make.

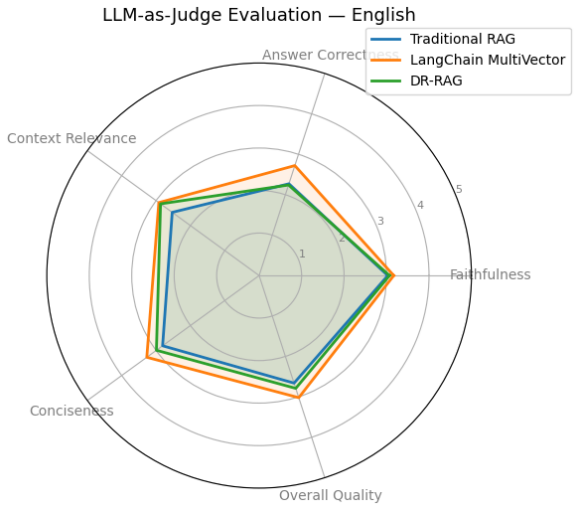


Figure 3: LLM-as-Judge radar chart for English QA. MultiVector leads on correctness; DR-RAG leads on faithfulness.

3.10. Comparative Analysis

The results tell a consistent story across all evaluation dimensions.

Where DR-RAG helps most. Urdu shows the biggest gains. QA pairs reduce the inflectional and paraphrastic variation that trips up chunk-only retrieval, which is exactly the morphological mismatch described in Section 2.3. The effect shows up in both retrieval (precision 60x higher than MultiVector) and generation (METEOR 38x above Traditional RAG). These are not narrow metric wins but broad quality improvements that hold across lexical, semantic, and factual dimensions.

English trade-offs. In English, strengths split across systems. MultiVector leads on lexical overlap and speed; DR-RAG leads on semantic alignment and faithfulness. The contrast suggests the dual index matters most when resources are scarce. When mature embeddings already bridge query-document gaps, the two approaches converge.

Speed and quality together. For Urdu, DR-RAG cuts generation time by 43% while improving quality. This happens because better retrieval produces shorter, tighter context for the generator. Getting both speed and accuracy improvements from the same architectural change is the clearest sign that the design is working as intended.

What the judge scores add. The qualitative evaluation reinforces the quantitative findings. DR-RAG is more faithful and more relevant in Urdu. In English it trades some conciseness for stronger grounding. Given that the primary use case is low-resource Urdu QA, that is a reasonable trade-off.

4. Conclusion

Retrieval failure in Urdu is not a general NLP problem but a specific structural one. The inflectional distance between query and document surface forms is wide enough that standard retrieval architectures, whether sparse or dense, consistently fail to establish reliable matches. DR-RAG addresses this by indexing document content in question form, creating representations that are syntactically and semantically closer to natural user queries. A confidence-aware fallback ensures the system remains robust when QA matching is uncertain, routing queries to chunk-based retrieval without significant overhead.

The results on Urdu UQA are substantial. METEOR improves by $38\times$, ROUGE-1 by 140%, and generation latency decreases by 43% relative to a traditional RAG pipeline. LLM-as-judge scores confirm stronger faithfulness and contextual relevance across all evaluated dimensions. Crucially, performance on English SQuAD 2.0 remains competitive with both baselines, demonstrating that these gains are additive rather than the result of optimising narrowly for Urdu at the expense of high-resource performance.

The central contribution of this work is not the architecture itself but the principle underlying it: that retrieval quality in low-resource languages depends on representational alignment between queries and indexed content, and that this alignment must be designed for rather than assumed. The improvements documented here emerge from linguistic reasoning about Urdu's specific properties rather than from increased model capacity or architectural complexity. This suggests the approach generalises naturally to other morphologically rich, low-resource languages that share similar retrieval failure modes, several of which are prominent across South Asia.

5. Limitations and Future Work

The current implementation has three notable limitations. The Urdu evaluation relies on only two indexed documents, a constraint imposed by the availability of suitable Urdu corpora at the time of experimentation. While the results are encouraging, validating whether these gains hold across larger and more topically diverse collections remains an important open question. Second, system output quality is evaluated using LLM-as-judge scoring via GPT-3.5 Turbo, without direct human evaluation of generated answers. While LLM-as-judge provides a scalable quality signal, human evaluation by native Urdu speakers remains an important next step to fully validate that these metrics reflect real-world perceived quality. Finally, the fallback threshold $\tau = 0.80$ was selected empirically

based on observed confidence score distributions, and has not been subjected to full sensitivity analysis. Retrieval confidence distributions in Urdu and other South Asian languages may differ substantially from English, suggesting that a learned, query-adaptive threshold policy would generalise more reliably across domains and linguistic settings.

Future work will extend evaluation to larger and more topically diverse Urdu document collections to validate that the observed gains generalise beyond the current small-scale index. Broader evaluation across additional low-resource South Asian languages sharing similar morphological and orthographic properties is also planned. The fallback threshold $\tau = 0.80$, currently selected empirically, will be subjected to systematic sensitivity analysis, with the goal of developing a learned, query-adaptive thresholding policy suited to latency-constrained deployment environments. Finally, DR-RAG will be benchmarked against recent RAG systems developed for Persian and Arabic, languages that share the same script family with Urdu, to assess cross-lingual transferability and position DR-RAG as a generalised solution across this script community.

Ethical Considerations

This study raises no significant ethical concerns. The datasets used, UQA and SQuAD 2.0, are publicly available, widely used in the research community, and contain no personally identifiable information. All experiments were conducted on locally hosted hardware, avoiding third-party data transmission and limiting environmental overhead. While DR-RAG improves factual grounding by anchoring generation in retrieved evidence, it does not eliminate the risk of bias inherited from pretrained models. This is a particular concern for low-resource South Asian languages, where the nature and extent of such biases are less thoroughly documented than for English, and warrants attention in any downstream deployment.

6. Bibliographical References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901.

Harrison Chase and LangChain Contributors. 2023. Langchain: Building applications with large language models. https://github.com/langchain-ai/langchain. GitHub Repository.

Michiel De Jong, Joris van Es, and Malvina Nissim. 2023. Fido: Fusion-in-decoder optimized for stronger retrieval-augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4571–4585, Singapore. Association for Computational Linguistics. Retrieval-Augmented Generation Optimization.

Layba Fiaz, Munief Hassan Tahir, Sana Shams, and Sarmad Hussain. 2025. Urdullama 1.0: Dataset curation, preprocessing, and evaluation in low-resource settings. *arXiv preprint arXiv:2502.16961*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2619–2630.

Omkar Khade, Shruti Jagdale, Abhishek Phaltankar, Gauri Takalikar, and Raviraj Joshi. 2025. Challenges in adapting multilingual LLMs to low-resource languages using LoRA PEFT tuning. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2021. Muril: Multilingual representations for indian languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4549–4564, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474.

Ziyang Li, Chen Liu, and Rui Zhang. 2025. Multihop-rag: Layered reasoning and evidence aggregation in retrieval-augmented generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Priyanka Mandikal and Raymond J. Mooney. 2024. Hybridir: A hybrid sparse and dense retrieval

framework for robust open-domain qa. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, et al. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2463–2473.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM Web Conference 2025 (TheWebConf 2025)*, Sydney, Australia. ACM.

Divyanshu Rajagopal, Aman Madaan, Sahil Gupta, Rajkumar Krishnan, Jiatong Chen, Alexander Zhang, Chris Callison-Burch, and Sameer Singh. 2023. Ragas: An evaluation framework for retrieval-augmented generation. arXiv preprint arXiv:2311.09514.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022. Colbert-x: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2762–2773.

Kurt Shuster, Jing Xu, Stephen Roller, and Jason Weston. 2024. Adaptive-rag: Reinforcement learning for adaptive retrieval in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, Bangkok, Thailand. Association for Computational Linguistics. Adaptive Retrieval Optimization for RAG.

Chi Su, Yu Wang, Zhi-Hong Deng, and Minlie Huang. 2024. Dragin: Dynamic retrieval-augmented generation via iterative neural querying. In *Proceedings of The 12th International Conference on Learning Representations (ICLR)*.

Hanna Tahir, Bameesha Bahim, and Asim Karim. 2024. Benchmarking multilingual models on urdu nlp tasks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, et al. 2023. Llama: Open and efficient foundation language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Pengcheng Yu, Zhengliang Liu, Qi Zhang, Yizhe Xie, and Yongchao Zhu. 2023. Generate-retrieve-read: Large language models are strong zero-shot retriever-readers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14318–14333, Singapore. Association for Computational Linguistics. Retrieval-Augmented Generation Framework.

Minghao Zhang, Yizhou Wang, and Jimmy Lin. 2025. Layered query retrieval: Structuring multi-hop reasoning in retrieval-augmented generation. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

7. Language Resource References

Traversaal AI. 2025. *Alif 1.0: Urdu-English Large Language Model*. Traversaal AI. Hugging Face, 1.0. PID https://huggingface.co/large-traversaal/Alif-1.0-8B-Instruct.

Samee Arif and Sualeha Farid and Awais Athar and Agha Ali Raza. 2024. *UQA: Corpus for Urdu Question Answering*. Urdu NLP Research Community. arXiv, 1.0. PID https://arxiv.org/abs/2405.01458.

UrduHack Contributors. 2021. *UrduHack: An NLP Toolkit for Urdu Language*. UrduHack. GitHub, 2.1. PID https://github.com/urduhack/urduhack.

Taha Farooq and Zunaira Khan. 2023. *LughaatNLP: Linguistic Toolkit for Low-Resource Urdu Text Processing*. Lughaat. GitHub, 1.0. PID https://github.com/lughaat/lughaatnlp.

Pranav Rajpurkar and Robin Jia and Percy Liang. 2018. *Know What You Don't Know: Unanswerable Questions for SQuAD*. Stanford NLP Group. arXiv. PID https://arxiv.org/abs/1806.03822.

Qdrant Team. 2023. *Qdrant: Vector Search Engine for the Next Generation of AI Applications*. Qdrant. Qdrant Technologies, 1.8. PID https://qdrant.tech.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. Improving multilingual text embedding with large language models. arXiv preprint arXiv:2402.05640.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Multilingual E5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05640*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 5776–5788.

Cross-Domain Evaluation of Transformer-Based Models for Punjabi Speech Emotion Recognition

Fatima Tu Zahra, Kulsoom Asim, Sandesh Kumar, Abdul Samad
Habib University, Dhanani School of Science and Engineering, Karachi, Pakistan
ft09259@st.habib.edu.pk, ka08051@st.habib.edu.pk,
sandesh.kumar@sse.habib.edu.pk, abdul.samad@sse.habib.edu.pk

Abstract

Speech Emotion Recognition (SER) is an important part of human–computer interaction, but most existing research focuses on high-resource languages, with very limited work on regional languages such as Punjabi. This paper focuses on detecting emotions from Punjabi speech using machine learning and deep learning techniques. We curated our own Punjabi speech emotion dataset using volunteer recordings and real-world sources, covering four emotion classes: angry, happy, sad, and neutral. The data was preprocessed for consistency and evaluated using a multi-strategy framework ($E1$–$E4$) to test domain generalization. Three models were evaluated: CNN-BiLSTM, ResNet-34, and the transformer-based Wav2Vec 2.0. Among these, the ResNet-34 model performed the best in the combined-domain strategy ($E4$), achieving a test accuracy of 96%. While cross-corpus evaluations ($E2$, $E3$) highlighted challenges in generalizing to neutral emotions, the model achieved perfect scores for happy and sad classes in $E4$. These results demonstrate the effectiveness of residual networks and combined-domain training for emotion recognition in low-resource languages and highlight the potential for further work on Punjabi SER.

Keywords: Speech emotion recognition, Punjabi speech, low-resource languages, deep learning, transformer models, Wav2Vec, ResNet, machine learning

1. Introduction

Punjabi is spoken by approximately 125 million people worldwide (SIL International, 2023) and is one of the major languages of South Asia. Despite its widespread use, publicly accessible resources for speech processing remain limited, particularly for Speech Emotion Recognition (SER). SER aims to identify human emotional states from vocal cues and has applications in mental health monitoring, human–computer interaction, call center analytics, and educational technologies (Ververidis & Kotropoulos, 2006; El Ayadi et al., 2011). However, most existing SER research focuses primarily on high-resource languages such as English and Mandarin, leaving low-resource languages like Punjabi comparatively underexplored.

A central challenge in low-resource SER is the scarcity of diverse and naturalistic datasets. Existing Punjabi resources, such as the RASA corpus (Varadhan et al., 2024), were originally developed for expressive text-to-speech (TTS) and consist of studio-recorded speech from only two speakers under controlled acoustic conditions. Although such datasets are valuable for synthesis tasks, they may not adequately reflect the acoustic variability present in real-world environments and may therefore limit model generalization across speakers and domains.

To address these limitations, we introduce a newly curated Punjabi SER dataset consisting of 893 labeled utterances across four emotion categories: happy, sad, angry, and neutral. The dataset combines volunteer recordings captured in both clean and noisy real-world settings with samples sourced from audiovisual media. This design increases speaker diversity and acoustic variability compared to studio-only corpora, providing a more realistic benchmark for emotion recognition in Punjabi speech.

In addition to dataset construction, we systematically investigate cross-dataset generalization in Punjabi SER. We design four experimental strategies to evaluate in-domain speaker-dependent evaluation on our curated dataset (E1), cross-domain transfer from RASA to curated speech (E2), reverse transfer from curated to RASA (E3), and combined-domain training using both datasets (E4). Through these controlled experiments, we analyze the impact of domain mismatch and speaker variation on model robustness. We further examine the effectiveness of transformer-based self-supervised speech models, particularly Wav2Vec 2.0, in both in-domain and cross-domain settings, while also comparing against ResNet-34 and CNN-BiLSTM baselines, with a focus on understanding generalization behavior under domain shift in low-resource South Asian languages.

2. Research Question

This study addresses the following research question: “Given an audio clip in Punjabi, how accurately can computational models detect the underlying emotional state across different speakers and recording conditions?”

Specifically, we evaluate model performance

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 59–67
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

on four emotion categories — angry, happy, sad, and neutral — under four experimental strategies: in-domain evaluation on our curated dataset (E1), cross-domain transfer from RASA to curated speech (E2), reverse transfer from curated speech to RASA (E3), and combined-domain training using both datasets (E4). This design allows us to assess both in-domain accuracy and robustness under domain and speaker variability.

3. Literature Review

Recent advances in Speech Emotion Recognition (SER) have significantly improved the ability of models to detect emotions in speech, even for low-resource languages like Punjabi. Modern deep learning approaches capture both temporal and contextual information in speech, which is crucial for modeling emotion dynamics. For example, Graph-LSTM networks treat speech frames as connected nodes, allowing models to track emotional changes across utterances (Li et al., 2023). Similarly, CNN–BiLSTM architectures, especially when combined with noise and spectral augmentations, demonstrate that combining convolutional feature extraction with sequential modeling enhances robustness under varying acoustic conditions (Barhoumi & Ben Ayed, 2023). However, most existing studies rely on controlled datasets with acted speech, limiting generalization to real-world, naturalistic recordings and low-resource languages (El Ayadi et al., 2011).

In Punjabi SER, early work by Singla & Singh (2022) introduced a semi-natural speech dataset of 985 utterances from Punjabi films and television shows, annotated for four emotions: happy, sad, angry, and neutral. They extracted prosodic, spectral, and wavelet features and tested classifiers such as Random Forest, SVM, Naïve Bayes, and Decision Trees, achieving a maximum accuracy of 61.65% with Random Forest. This study highlighted the importance of feature engineering for low-resource language SER. Later, Singla et al. (2024) applied deep learning, using CNNs on a larger dataset of 9,000 utterances from films, web series, and social media. Preprocessing, segmentation, and augmentation addressed class imbalance, and their model achieved 69% accuracy, demonstrating the advantage of high-level spectral features and deep architectures. Chaitanya Singla and Sukhdev Singh (Singla & Singh, 2022) also released PEMO, a large-scale natural Punjabi speech emotion dataset of 22,000 utterances with strict label agreement, capturing real-life emotional expressions and providing a valuable benchmark for future models. Kaur & Singh (2022) further emphasized the role of feature selection and extraction in improving Punjabi SER performance using CNNs.

Despite these contributions, critical gaps remain. Many datasets, including the 9,000-utterance corpus (Singla et al., 2024) and PEMO (Singla & Singh, 2022), are not publicly accessible, limiting reproducibility and comparative evaluation. Additionally, prior studies rarely evaluate model robustness under domain shift or cross-dataset scenarios, leaving the generalizability of models to naturalistic speech untested.

4. Dataset

4.1. Source and Collection

Our Punjabi Speech Emotion Recognition (SER) dataset draws from two complementary sources: volunteer recordings and publicly available audiovisual media. This dual-source design was adopted to maximise acoustic and speaker diversity, combining controlled scripted utterances with naturalistic expressive speech extracted from real-world content.

For the volunteer component, ten participants contributed recordings spanning four target emotion categories: anger, happiness, sadness, and neutral. Participants ranged in age from 21 to 60 years, with an equal gender split of five male and five female speakers. Seven participants were native Punjabi speakers; the remaining three were fluent non-native speakers. This demographic composition is summarised in Table 1.

Each participant was provided with five prewritten scripts, each containing ten sentences per emotion class (40 sentences per script), yielding a pool of scripted utterances. Recordings were conducted in both quiet and acoustically noisy environments to increase domain diversity. Individual utterance durations ranged from approximately 2 to 7 seconds depending on speaker fluency.

All audio files were standardised to mono 16-bit PCM WAV format at a sampling rate of 16 kHz. Approximately half of the recordings were denoised and amplitude-normalised in post-processing; the remainder were retained in their original acoustic condition to preserve variability. The neutral-category samples were sourced from the Shrutilipi ASR corpus (Bhogale et al., 2022), a publicly available corpus of transcribed Indian language speech developed for automatic speech recognition, and were used to supplement volunteer recordings for this affectively ambiguous class. The dataset comprises read news recordings from All India Radio, with audio provided in 16 kHz WAV format and aligned transcriptions in fairseq format.

The audiovisual component was assembled by extracting emotionally labelled utterances from Indian Punjabi films and television programmes,

providing naturally occurring speech with diverse speaker identities, recording conditions, and expressive styles not present in scripted volunteer data. While all audiovisual sources reflect Indian Punjabi, the volunteer component provides Pakistani Punjabi representation — all ten volunteer speakers were Pakistani Punjabi speakers. Although phonetic and prosodic differences between Indian and Pakistani Punjabi are not extensive, cross-border dialectal coverage remains uneven across the two components, and expanding audiovisual representation to Pakistani Punjabi content is an avenue for future work.

The final dataset comprises 893 labelled utterances used for all experiments reported in this paper.

4.2. Speaker Demographics

Table 1 summarises the demographic characteristics of the volunteer participant pool.

Table 1: Volunteer Speaker Demographics

Characteristic	Value
Total Speakers	10
Male Participants	5 (50%)
Female Participants	5 (50%)
Age Range	21–60 years
Native Punjabi Speakers	7 (70%)
Fluent Non-Native Speakers	3 (30%)
Utterance Duration	2–7 seconds

4.3. Class Distribution

The dataset covers four emotion categories: angry, happy, neutral, and sad. Table 2 presents the per-class sample counts broken down by source. Neutral samples were sourced exclusively from the Shrutilipi corpus (Bhogale et al., 2022) rather than audiovisual media, as neutral speech is rarely unambiguously expressed in dramatic contexts. The resulting class distribution is imbalanced, with sad utterances constituting the largest class (327) and happy the smallest (160). Such imbalance is an inherent property of naturalistic collection methodologies and introduces distributional priors that may interact with domain shift under cross-domain conditions.

4.4. RASA Dataset

The RASA dataset is a multilingual expressive Text-to-Speech (TTS) corpus (Varadhan et al., 2024) developed to support speech synthesis research across 22 Indian languages, including Punjabi. Each language subset features one male and one female speaker recorded under controlled, anechoic studio conditions. Audio is stored in mono WAV format at 48 kHz, with utterance durations ranging from 3 to 15 seconds. Despite its synthesis-oriented origins, the expressive nature of RASA recordings — spanning multiple emotion categories with consistent annotation — renders it suitable as a complementary SER resource, particularly as a controlled-domain counterpart to our naturalistic curated dataset.

Table 2: Emotional Class Distribution by Data Source

Emotion	Volunteer	AV Media	Shrutilipi*	Total
Angry	141	100	0	241
Happy	60	100	0	160
Neutral	0	0	165	165
Sad	233	94	0	327
Total	**434**	**294**	**165**	**893**

* Neutral samples sourced exclusively from the Shrutilipi corpus (Bhogale et al., 2022).

For this study, the Punjabi subset of RASA was used, covering the same four emotion classes as our curated dataset: angry, happy, sad, and neutral. RASA provides a predefined train–test split comprising 8,672 training samples and 962 test samples. The neutral class is heavily overrepresented, accounting for 6,039 of the 8,672 training samples. To mitigate this imbalance, two strategies were applied depending on the model architecture: for ResNet34, neutral training samples were capped at 2,000; for CNN-BiLSTM and Wav2Vec 2.0, class-weighted loss functions were used during training. The test partition was left unchanged in all cases to ensure fair evaluation.

Table 3 presents the class-level breakdown of the RASA Punjabi subset.

Table 3: RASA Punjabi Subset: Class Distribution

Emotion	Training Set	Test Set
Angry	861	95
Happy	915	102
Sad	857	95
Neutral	6,039	670
Total	**8,672**	**962**

The acoustic divergence between RASA's studio-controlled recordings and the naturalistic, variable-condition speech in our curated dataset provides the primary motivation for the cross-domain experimental framework. This contrast allows us to isolate the effect of domain mismatch from confounds such as language or emotion label differences, yielding a controlled setting for evaluating model generalisation.

5. Methodology

5.1. Pre-processing

All audio files were resampled to 16 kHz to ensure a uniform sampling rate across both corpora (Baevski et al., 2020). Each utterance was normalised to a fixed length of 16,000 samples (1 second at 16 kHz) via zero-padding for shorter clips and centre-cropping for longer ones. Processed files were structured into a Hugging Face `Dataset` object to standardise splitting, batching, and feature extraction across all experimental conditions.

5.2. Feature Extraction

Feature extraction followed a multi-representation strategy to capture complementary acoustic dimensions of emotional speech. The primary feature set comprised Mel-Frequency Cepstral Coefficients (MFCCs), which encode the short-term power spectrum of speech on a perceptually motivated scale (Gourisaria et al., 2024). Four additional feature families were extracted: chroma features, mel spectrograms, spectral contrast, and tonnetz features. Concatenating these descriptors into a unified feature vector provides complementary spectral, temporal, and harmonic information, improving discriminability across emotion classes that share prosodic surface characteristics.

5.3. Dataset Splitting

The splitting strategy varied by experimental strategy. For experiments using the curated dataset, an 80/20 stratified split was applied. The split was not enforced to be speaker-independent, and E1 results should be interpreted with this caveat in mind. For experiments involving RASA, no splitting was performed; the dataset was used in its entirety for either training or testing depending on the experimental strategy. Complete details of each configuration are presented in Table 4.

5.4. Model Architectures

Three architectures were evaluated across two model families. CNN-BiLSTM combines a convolutional front-end for local spectro-temporal feature extraction with a bidirectional LSTM for sequential modelling. ResNet34 applies 34-layer residual learning with skip connections; training used pitch shifting, time stretching, and temporal shifting augmentation, Adam optimisation with a discriminative learning rate (10^{-5} to 10^{-4}), and early stopping on validation loss. Wav2Vec 2.0 (`wav2vec2-xls-r-300m`) is a 300M-parameter transformer pretrained on large-scale multilingual speech (Baevski et al., 2020), fine-tuned here for four-class Punjabi emotion classification. All models were evaluated on accuracy, macro F1-score, precision, and recall.

5.5. Experimental Strategies

Four strategies were designed to evaluate in-domain performance, cross-domain transfer, and generalisation under domain shift — a dimension largely absent from prior Punjabi SER studies. Table 4 summarises each configuration, where "Curated" refers to our naturalistic Punjabi SER dataset.

Table 4: Experimental Training and Testing Strategies

Strategy	Training Set	Testing Set
E1	Curated	Curated
E2	RASA	Curated
E3	Curated	RASA
E4	RASA + Curated	Curated

E1 establishes in-domain baselines using the curated dataset for both training and testing. E2 and E3 probe cross-domain transfer in both directions between studio and naturalistic speech. E4 utilises a combined training set of both corpora to assess whether heterogeneous data can recover performance lost under domain shift.

6. Results

In this section, we evaluate the performance of the proposed models across the four experimental strategies (E1–E4) and analyze their effectiveness in recognizing emotions in Punjabi speech under both in-domain and cross-domain conditions.

6.1. CNN-BiLSTM

Table 5: CNN-BiLSTM Performance Across Experimental Strategies

Strategy	Macro Acc. (%)	Angry (%)	Happy (%)	Sad (%)	Neutral (%)
E1	85.39	100.0	80.0	66.7	93.94
E2	37.7	36.2	42.7	24.2	47.5
E3	34.0	54.6	0.0	29.4	51.8
E4	83.7	72.9	90.0	81.2	90.6

CNN-BiLSTM was evaluated for four-class emotion classification across all experimental strategies. Table 5 presents macro accuracy and per-emotion performance.

In the in-domain setting (E1), the model achieved high performance across emotions, with accuracies of 100.0% (angry), 80.0% (happy), 66.7% (sad), and 93.94% (neutral), resulting in a macro accuracy of 85.39%. This indicates that the model effectively

captures emotional patterns when training and testing distributions are aligned.

Cross-domain evaluation (E2) revealed a substantial drop in performance, with macro accuracy of 37.7%. While neutral achieved moderate recognition (47.5%), sad was poorly classified (24.2%), highlighting sensitivity to acoustic variability and domain mismatch between training and testing datasets.

Reverse transfer (E3) further exposed generalization limitations, yielding 34.0% macro accuracy. The model collapsed on happy (0.0%) and underperformed on sad (29.4%), demonstrating strong dataset-specific bias and insufficient domain-invariant learning.

Combined-domain training (E4) improved robustness considerably, achieving 83.7% macro accuracy. Performance became more balanced across emotions: angry (72.9%), happy (90.0%), sad (81.2%), and neutral (90.6%). Although slightly lower than the best in-domain result, this strategy substantially mitigated cross-domain degradation, demonstrating that heterogeneous training data enhances generalization in emotion recognition across diverse acoustic conditions.

6.2. ResNet34

Table 6: ResNet34 Performance Across Experimental Strategies

Strategy	Macro Acc. (%)	Happy (%)	Anger (%)	Sad (%)	Neutral (%)
E1	92.00	88.00	89.00	94.00	94.00
E2	37.36	27.33	62.92	40.80	1.85
E3	43.97	6.86	57.89	53.88	0.00
E4	96.00	100.00	89.58	100.00	93.93

ResNet-34 was evaluated across all four experimental strategies for four-class emotion classification. To mitigate overfitting on the curated Punjabi samples, we employed pitch shifting, time stretching, and time shifting augmentations. Optimization was conducted via the Adam optimizer using a discriminative learning rate slice (10^{-5} to 10^{-4}), with an Early Stopping mechanism to ensure convergence based on validation loss.

As shown in Table 6, the model achieved its peak performance under Strategy E4 (combined-domain training), reaching an overall accuracy of 96%. The model reached 100% accuracy for both the Happy and Sad categories, indicating that these emotions exhibit very clearly distinguishable acoustic patterns in the Punjabi corpus. A slight performance decrease was noted for Neutral (93.93%), while Anger proved challenging, at 89%. This relative drop in Anger likely reflects the acoustic overlap between high-intensity vocalizations and the noise present in the curated data.

The model exhibited its lowest performance in strategy E2, achieving an overall accuracy of 37%. In particular, it demonstrated substantial difficulty in correctly classifying neutral utterances. A comparable decrease in accuracy was observed in strategy E3; while slightly better at 43%, it failed to classify neutral entirely. Training details are provided in Table 7.

Table 7: ResNet34 Training Parameters

Parameter	Value
Model Type	ResNet34
Epochs	20
Batch Size	16
Optimizer	Adam
Learning Rate	Slice (10^{-5} to 10^{-4})
Loss Function	Flat Cross-Entropy

6.3. Wav2Vec 2.0

Table 8: Wav2Vec 2.0 Performance Across Experimental Strategies

Strategy	Macro Acc. (%)	Angry (%)	Happy (%)	Sad (%)	Neutral (%)
E1	92.48	100.0	87.0	86.0	100.0
E2	41.3	46.7	43.3	54.9	20.4
E3	25.0	0.0	0.0	100.0	0.0
E4	82.8	81.2	76.7	73.4	100.0

Wav2Vec 2.0 was fine-tuned for four-class emotion classification using contextualized speech representations derived from self-supervised pretraining. Table 8 reports macro accuracy along with per-emotion performance for all experimental strategies.

In the in-domain setting (E1), the model achieved strong and balanced performance across all emotions, with accuracies of 100.0% (angry), 87.0% (happy), 86.0% (sad), and 100.0% (neutral), resulting in a macro accuracy of 92.48%. This demonstrates that transformer-based representations effectively capture emotional characteristics when training and testing distributions are aligned.

Under distributional shift, performance degraded substantially. In E2, cross-domain transfer from RASA to curated speech yielded 41.3% macro accuracy, with moderate performance on sad (54.9%) but poor recognition of neutral (20.4%), highlighting the model's sensitivity to acoustic and recording variability.

Reverse transfer (E3) further exposed generalization limitations, with a macro accuracy of 25.0%. The model collapsed toward the sad class (100%) while failing to classify the remaining emotions, indicating strong dataset-specific bias rather than domain-invariant learning.

Combined-domain training (E4) improved robustness considerably, achieving 82.8% macro accuracy. Performance became more balanced, with

notable improvements in angry (81.2%) and happy (76.7%), and perfect recognition of neutral (100%). Although slightly lower than the best in-domain result, this strategy substantially outperformed pure cross-domain transfer, demonstrating that exposure to heterogeneous training data mitigates domain shift and enhances generalization.

7. Discussion

7.1. In-Domain Performance and the Value of Naturalistic Data

The strong in-domain results achieved across all three architectures in E1 — Wav2Vec 2.0 at 92.48% macro accuracy, ResNet34 at 92.00%, and CNN-BiLSTM at 85.39% — establish that naturalistic Punjabi speech data, even at modest scale, supports effective emotion recognition when training and test distributions are aligned. These results indicate that data ecological validity — the degree to which a dataset reflects real-world speech conditions — contributes substantially to in-domain model performance. The competitive performance of ResNet34 relative to Wav2Vec 2.0 further suggests that convolutional architectures with residual connections are capable of extracting stable spectro-temporal features from short naturalistic utterances, and that the representational advantage of transformer-based models is most pronounced under conditions of extreme data scarcity rather than when a well-distributed naturalistic corpus is available.

7.2. Cross-Domain Transfer and the Studio-to-Naturalistic Gap

The most significant finding of this study is the severity and consistency of performance degradation under cross-domain conditions. In both E2 (RASA $\rightarrow$ Curated) and E3 (Curated $\rightarrow$ RASA), all three models experienced dramatic reductions in macro accuracy. This degradation is attributable to two compounding factors. The primary source is the fundamental acoustic domain mismatch between RASA's controlled, anechoic studio recordings and the naturalistic recordings in our curated dataset, which encompass ambient noise, variable microphone conditions, and spontaneous emotional expression drawn from real-world audiovisual content. The second contributing factor is class imbalance, which amplifies cross-domain collapse by introducing strong distributional priors: the neutral class dominates RASA at 6,039 of 8,672 training samples, while *sad* constitutes the majority class in our curated set at 327 of 893 samples. Under domain shift, when cross-domain acoustic features become unreliable, models exploit these priors and converge toward the majority class of their training distribution — a well-documented failure mode in cross-corpus SER. The complete convergence of Wav2Vec 2.0 toward the *sad* class in E3 — achieving 100% sad recall while failing entirely on all other classes — exemplifies this mechanism precisely.

The directional asymmetry between E2 and E3 provides further insight: transfer from RASA to the curated set consistently outperforms the reverse direction, indicating that pretraining on larger, acoustically cleaner data confers a modest but measurable generalisation benefit. Nevertheless, this advantage remains insufficient to overcome the acoustic divergence between studio and naturalistic domains.

7.3. Combined-Domain Training as a Practical Strategy

E4 substantially recovers performance across all models, with ResNet34 achieving 96.00% and CNN-BiLSTM reaching 83.7%, indicating increased robustness to domain variation when heterogeneous training data is available. These results suggest that combining available controlled corpora with modest naturalistic data is more effective than relying on either source in isolation. The divergence between ResNet34 (96.00%) and Wav2Vec 2.0 (82.8%) in E4 is explained by the fine-tuning dynamics of each architecture. ResNet34 is trained from scratch on the combined corpus with task-specific augmentation — pitch shifting, time stretching, and temporal shifting — enabling full adaptation to the combined distribution. Wav2Vec 2.0 enters fine-tuning with deep representational priors from large-scale multilingual pretraining, which may not fully align with Punjabi emotional speech acoustics; when the fine-tuning corpus remains small relative to pretraining scale, these priors can hinder full adaptation to the target domain.

7.4. Broader Implications for South Asian SER

These findings collectively establish that acoustic domain match is a key determinant of model performance in Punjabi SER, in this setting often exerting greater influence than architectural sophistication or pretraining scale. Evaluation benchmarks that report exclusively in-domain performance may overstate model generalisability — a concern of particular consequence when studio corpora constitute the primary available resource for South Asian languages. The cross-domain evaluation framework introduced in this study provides a more diagnostically informative paradigm that can complement standard in-domain assessment in future work. Furthermore, the E4 results demonstrate that small naturalistic corpora provide substantial gains

when combined with larger controlled datasets — a tractable data collection strategy for South Asian language communities operating under resource constraints. Given the phonetic and prosodic proximity of Punjabi to Hindi and Urdu, the cross-domain evaluation framework introduced here may generalise naturally to these related languages, and future work exploring multilingual transfer across South Asian languages — including whether Hindi or Urdu training data can improve Punjabi SER performance — would position this line of research within a broader coordinated regional effort.

8. Error Analysis

8.1. Emotion-Level Confusion Patterns

Examination of per-class performance reveals consistent patterns across models and strategies that macro accuracy alone does not capture. In the in-domain setting (E1), angry achieved perfect precision and recall (100%) across both CNN-BiLSTM and Wav2Vec 2.0, reflecting its acoustically distinctive profile — elevated pitch, high energy, and rapid articulation. ResNet34 similarly performed strongly in E1, with the only notable errors being 3 angry utterances misclassified as sad and 3 happy utterances misclassified as angry, consistent with acoustic overlap between high-intensity emotions. The clearest in-domain weakness was sad: CNN-BiLSTM recorded only 66.67% recall despite 100% precision, misclassifying 5 sad utterances as neutral and 3 as happy. This confusion is consistent with the acoustic proximity between subdued sadness and neutral speech, both characterised by reduced pitch range, lower energy, and slower speaking rates. Wav2Vec 2.0 showed a complementary pattern, with happy (87% recall, 77% precision) as its primary weakness — predominantly misclassified as neutral — suggesting that low-arousal happy expressions in naturalistic Punjabi share prosodic characteristics with neutral utterances.

Under cross-domain conditions, these confusions intensified severely. In E2 (RASA → Curated), CNN-BiLSTM's sad recall collapsed to 24.2%, while Wav2Vec 2.0 showed the inverse pattern: highest sad recall (54.9%) but near-collapse on neutral (20.4%). ResNet34 under E2 exhibited a striking neutral collapse — of 162 neutral test instances, 156 were misclassified as sad, mirroring the majority-class prior effect observed in Wav2Vec 2.0 under E3. In E3 (Curated → RASA), Wav2Vec 2.0 assigned sad to all test instances — achieving 100% sad recall and 0% on all other classes — a complete distributional collapse driven by the majority-class prior of the curated training set interacting with the acoustic divergence of studio test conditions. CNN-BiLSTM in E3 exhibited a comparatively distributed failure, with happy collapsing entirely (0%) while angry (54.6%) and neutral (51.8%) retained partial recognition. ResNet34 under E3 showed a distinct failure mode: sad recalled 0%, with all sad instances redistributed to angry (20) and neutral (75), while angry and neutral retained partial recognition.

Combined-domain training (E4) substantially restored class balance across all three models. CNN-BiLSTM exceeded 72% recall on all four classes, with happy (90.0%) and neutral (90.6%) recovering most strongly. Wav2Vec 2.0's sad (73.4%) remained its relative weak point while neutral achieved perfect recognition (100%) — a complementary profile to CNN-BiLSTM that points toward ensemble approaches as a promising direction for further gains. ResNet34 in E4 achieved the most balanced recovery of all models, with near-perfect classification across all four classes and minimal inter-class confusion, consistent with its 96% macro accuracy.

8.2. Cross-Model Error Consistency

A consistent finding across all three models and all strategies is that angry is the most robustly recognised emotion and that neutral recognition is the most fragile under domain shift. The happy–sad pair remains the most frequently confused under in-domain conditions, with both emotions capable of manifesting with moderate pitch and tempo in naturalistic Punjabi depending on speaker register — a fundamental acoustic ambiguity that warrants targeted dataset design in future work. Table 9 summarises the dominant misclassification patterns across all models and strategies.

Table 9: Dominant Misclassification Patterns Across Models and Strategies

Strategy	Model	Primary Confusion	Secondary Confusion
E1	CNN-BiLSTM	Sad → Neutral	Happy → Neutral
E1	ResNet34	Angry → Sad	Happy → Angry
E1	Wav2Vec 2.0	Happy → Neutral	Sad → Happy
E2	CNN-BiLSTM	Sad → Neutral	Angry → Neutral
E2	ResNet34	Neutral → Sad	Happy → Angry
E2	Wav2Vec 2.0	Neutral → Sad	Angry → Sad
E3	CNN-BiLSTM	Happy → Neutral	Sad → Neutral
E3	ResNet34	Sad → Neutral	Happy → Angry
E3	Wav2Vec 2.0	All → Sad	—
E4	CNN-BiLSTM	Angry → Sad	Sad → Neutral
E4	ResNet34	Angry → Sad	—
E4	Wav2Vec 2.0	Sad → Happy	Happy → Sad

9. Comparative Analysis

Table 10 situates our results within the existing Punjabi SER literature. Direct numerical comparison across studies is complicated by differences in dataset composition, emotion label sets, and evaluation methodology; however, the table estab-

lishes approximate performance trajectories across successive approaches.

Table 10: Comparison with Prior Punjabi SER Work

Study	Dataset	Best Model	Accuracy (%)
Singla & Singh (2022)	985 utterances (film/TV)	Random Forest	61.65
Singla et al. (2024)	9,000 utterances (film/web)	CNN	69.00
Kaur & Singh (2022)	900 utterances (recorded)	CNN	81.00
This work (E1)	**893 naturalistic**	**Wav2Vec 2.0**	**92.48**
This work (E4)	**RASA + 893 naturalistic**	**ResNet34**	**96.00**

Our in-domain result of 92.48% (Wav2Vec 2.0, E1) represents a substantial improvement over the prior state-of-the-art for Punjabi SER. Singla & Singh (2022) achieved 61.65% using classical feature engineering with Random Forest on semi-natural film speech, while Singla et al. (2024) advanced this to 69% using CNNs on a considerably larger 9,000-utterance corpus. Kaur & Singh (2022) further advanced this to 81.00% using a 1D CNN with LASSO feature selection on a purpose-recorded 900-utterance corpus, demonstrating the benefit of dedicated data collection over media-sourced datasets. Our results demonstrate that transformer-based self-supervised representations close this gap significantly, achieving over 23 percentage points of improvement relative to Singla et al. (2024) on a smaller but more acoustically diverse dataset. The combined-domain result of 96.00% (ResNet34, E4) further extends this margin, suggesting that data heterogeneity is as consequential as model architecture in advancing Punjabi SER performance.

Critically, prior Punjabi SER studies evaluate exclusively in-domain, leaving cross-domain generalisation entirely uncharacterised. This study introduces the first systematic cross-domain evaluation framework for Punjabi SER, establishing baseline cross-domain performance figures — E2: 37.7–41.3%, E3: 25.0–34.0% — that reveal the brittleness of current approaches under domain shift and provide a reproducible benchmark against which future work can be measured.

10. Conclusion

This paper presented a systematic cross-domain evaluation of deep learning and transformer-based models for Punjabi Speech Emotion Recognition. We introduced a curated dataset of 893 naturalistic utterances and evaluated CNN-BiLSTM, ResNet34, and Wav2Vec 2.0 across four experimental strategies — in-domain (E1), cross-domain transfer (E2), reverse transfer (E3), and combined-domain training (E4) — characterising model performance under both aligned and mismatched distribution conditions.

Three principal findings emerge. First, Wav2Vec 2.0 achieves 92.48% macro accuracy in-domain, substantially advancing the prior Punjabi SER state-of-the-art. Second, cross-domain transfer degrades severely across all models and both transfer directions, demonstrating that in-domain evaluation alone is insufficient to assess practical model utility. Third, combined-domain training recovers performance effectively — ResNet34 reaching 96.00% — establishing data heterogeneity as the most practical development strategy for low-resource Punjabi SER. This work further contributes the first cross-domain evaluation framework for Punjabi SER, providing reproducible baselines and motivating evaluation protocols that assess domain generalisation alongside in-domain accuracy. Future work will prioritise dialectal dataset expansion to cover Pakistani Punjabi audiovisual content, domain adaptation techniques for studio-to-naturalistic transfer, multilingual transfer learning across South Asian languages including Hindi and Urdu, and extension to dimensional emotion recognition.

Ethics and Limitations Statement

All volunteer participants provided informed consent prior to recording. Participants were informed of the research purpose and intended use of their recordings, and participation was voluntary. Audiovisual media samples were used solely for non-commercial academic research. No personally identifiable information beyond basic demographic characteristics is retained. Speech emotion recognition systems carry inherent risks of misuse in surveillance and non-consensual affect monitoring; the authors do not endorse such applications, and the resources developed in this work are intended strictly for research use under appropriate ethical oversight.

This work has several limitations. First, the curated dataset comprises 893 utterances from a small volunteer pool of ten speakers, and the 80/20 train-test split was not enforced to be speaker-independent; E1 results should therefore be interpreted with caution as they may reflect some degree of speaker overlap. Second, all audiovisual media sources reflect Indian Punjabi, while volunteer speakers represent Pakistani Punjabi; cross-border dialectal coverage therefore remains uneven, and generalisation to Pakistani Punjabi audiovisual content is untested. Third, the dataset does not represent the full dialectal diversity of Punjabi as spoken across India and Pakistan, and findings may not generalise uniformly across all regional varieties. Finally, the cross-domain results establish that models trained on studio-controlled corpora generalise poorly to naturalistic speech, suggesting that the high in-domain accuracies reported here and in prior work overstate practical deployability.

Acknowledgements

The authors sincerely thank all volunteer participants for contributing their recordings to this study. We also acknowledge the providers of publicly available datasets used in this work, whose resources made this research possible. Their contributions are greatly appreciated.

Data and Code Availability Statement

The dataset is publicly available on Kaggle. The code is available on this repository.

11. Bibliographical References

[1] Aggarwal, S., Singh, P. & Sharma, K. (2023). A survey on speech emotion recognition: Datasets, features, and classification methods. *Journal of Intelligent Systems*, 33(1), pp. 203–220.

[2] Baevski, A., Zhou, H., Mohamed, A. & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020)*.

[3] Barhoumi, C. & Ben Ayed, Y. (2023). Real-Time Speech Emotion Recognition Using Deep Learning and Data Augmentation. *Research Square* [Preprint]. doi: 10.21203/rs.3.rs-2874039/v1.

[4] Bhogale, K. S., Jain, N., Raman, A., Agarwal, S., Chhimwal, A., Bansal, D. & Jyothi, P. (2022). Effectiveness of Mining Audio and Text Pairs from Public Data for Improving ASR Systems for Low-Resource Languages. *arXiv preprint arXiv:2208.12666*.

[5] El Ayadi, M., Kamel, M. S. & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3), pp. 572–587.

[6] Gourisaria, M. K., Agrawal, R., Sahni, M. & Singh, P. K. (2024). Comparative analysis of audio classification with MFCC and STFT features using machine learning techniques. *Discover Internet of Things*, 4(1).

[7] Kaur, K. & Singh, P. (2022). Impact of Feature Extraction and Feature Selection Algorithms on Punjabi Speech Emotion Recognition Using Convolutional Neural Network. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21(5), Article 100.

[8] Li, Y., Wang, Y., Yang, X. & Im, S. (2023). Speech emotion recognition based on Graph-LSTM neural network. *EURASIP Journal on Audio, Speech, and Music Processing*, 2023(40).

[9] SIL International. (2023). Punjabi Language Profile. *Ethnologue*. [Online]. Available: https://www.ethnologue.com/language/pan

[10] Singla, C. & Singh, S. (2022). Punjabi Speech Emotion Recognition using Prosodic, Spectral and Wavelet Features. In *Proceedings of the 2022 10th IEEE International Conference on Emerging Trends in Engineering & Technology Signal and Information Processing (ICETET-SIP-22)*, pp. 1–6.

[11] Singla, C., Singh, S., Sharma, P., Mittal, N. & Gared, F. (2024). Emotion Recognition for Human–Computer Interaction Using High-Level Descriptors. *Scientific Reports*, 14, p. 12122.

[12] Varadhan, P. S., Sankar, A., Raju, G. & Khapra, M. M. (2024). Rasa: Building Expressive Speech Synthesis Systems for Indian Languages in Low-resource Settings. In *Proceedings of INTERSPEECH 2024*.

[13] Ververidis, D. & Kotropoulos, C. (2006). Emotional speech recognition: Resources, features and methods. *Speech Communication*, 48(9), pp. 1162–1181.

12. Language Resource References

[1] Singla, C. & Singh, S. (2022). PEMO: A New Validated Dataset for Punjabi Speech Emotion Detection. *International Journal on Recent and Innovation Trends in Computing and Communication*, 10(10), pp. 52–57. ISLRN 2321-8169.

SiPaKosa: A Comprehensive Corpus of Canonical and Classical Buddhist Texts in Sinhala and Pali

Ranidu Gurursinghe♠ and Nevidu Jayatilleke♣
♠School of Computing, Informatics Institute of Technology, Sri Lanka
♣Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka
ranidu.20222198@iit.ac.lk, nevidu.25@cse.mrt.ac.lk

Abstract

SiPaKosa is a comprehensive corpus of Sinhala and Pali doctrinal texts comprising approximately 786K sentences and 9.25M words, incorporating 16 copyright-cleared historical Buddhist documents alongside the complete web-scraped Tripiṭaka canonical texts. The corpus was created through high-quality OCR using `Google Document AI` on historical manuscripts, combined with systematic web scraping of canonical repositories, followed by rigorous quality control and metadata annotation. The corpus is organised into language-specific subcorpora: Sinhala and Mixed Sinhala-Pali. We evaluate the performance of language models using ten pretrained models, with perplexity scores ranging from 1.09 to 189.67 on our corpus. This analysis shows that proprietary models significantly outperform open-source alternatives by factors of three to six times. This corpus supports the pretraining of domain-adapted language models, facilitates historical language analysis, and aids in the development of information retrieval systems for Buddhist scholarship while preserving Sinhala cultural heritage.

Keywords: Sinhala NLP, Pali, Low-resource Language, Digital Humanities

1. Introduction

සබ්බදානං ධම්මදානං ජිනාති[1] (Thera and Rao, 1954), highlights that sharing knowledge and truth is considered the highest form of generosity because it provides the tools for others to free themselves. While the importance of the universal sharing of knowledge is highlighted, the field of computational linguistics remains significantly under-resourced in terms of structured religious corpora (Hutchinson, 2024).

The Sinhala language is part of the Indo-Aryan branch of the Indo-European language family with a rich and diverse literary heritage that has developed over several millennia. Its origins can be traced back to between the 3rd and 2nd centuries BCE. It is the primary language of the Sinhalese people, who represent the largest ethnic group in Sri Lanka, and it is recognised as the first language (L1) for approximately 16 million people (De Silva, 2026). Furthermore, Sinhala is classified as a lower-resourced language (Category 01) according to the criteria presented by Ranathunga and de Silva (2022).

Pali, which was historically a language commonly used by monks of different nations for communication (Zigmond, 2023), is considered a dead language despite being widely studied because it is the language of early Buddhist scriptures (Knauth and Alfter, 2014). In terms of data resources, Pali is also a category 1 low-resource language according to the criteria presented by Ranathunga and de Silva (2022). In this work, we will focus on Pali written in the Sinhala script (the Sinhala script is also used for writing Pali and Sanskrit literature in Sri Lanka (Gair, 1996)), rather than the more commonly known Devanagari script, which presents additional challenges.

The digitisation of historical texts presents several significant challenges. These include the difficulty of OCR on degraded copies, the inconsistency of spelling and orthography over the centuries (Jayatilleke and de Silva, 2025a), and the limited research that has been conducted on OCR for scanned documents and images featuring text written in the Sinhala script (De Silva, 2026).

We introduce SiPaKosa[2], a large-scale corpus combining classical Sinhala Buddhist texts from historical archives with complete canonical scriptures from web sources. Our contributions include: (1) a corpus of 786,839 sentences (∼9.25M words) from 16 historical documents and complete Tripiṭaka texts, (2) comprehensive metadata and language classification, (4) language-separated subcorpora Sinhala (465,539 sentences, 5.42M tokens), Pali (495 sentences, 3.2K tokens), and Mixed Sinhala-Pali (320,805 sentences, 3.83M tokens), and (5) evaluations showing 3-6 times gaps between proprietary and open-source models in terms of perplexity scores.

2. Related Work

2.1. The Bible in 100 languages

A multilingual Bible Corpus was introduced by Christodouloupoulos and Steedman (2015), comprising translations into 100 languages. The

[1] **English Meaning:** *The gift of dhamma (truth) conquers all other gifts; it truly excels and surpasses them all.*
Sinhala to IPA Transliteration: sʌbbʌðɑ:nʌŋ ðʰʌmmʌðɑ:nʌŋ dʒɪnɑ:θɪ.

[2] https://github.com/RanxduG/SiPaKosa-Dataset

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 68–84
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

corpus was constructed by web scraping publicly available Bible translations, automatic sentence alignment using the inherent verse-level structure of Biblical texts, and finally, quality filtering based on translation completeness and linguistic coverage. The resulting corpus structure preserves both the original hierarchical metadata (book names, chapter numbers, verse identifiers) and linguistic annotations (sentence boundaries, tokenisation), facilitating diverse research applications from machine translation to typological studies.

While serving different theological traditions, both the Bible and the Buddhist corpora share critical characteristics: these texts hold canonical status within their respective religious traditions (the Bible for Christianity, the Tripiṭaka for Buddhism). This status necessitates not only linguistic accuracy but also the preservation of religious meanings and theological concepts during digitisation. Furthermore, both face challenges related to translation and OCR across diverse linguistic families, along with the responsibility to safeguard sacred texts through digital means while ensuring accuracy for researchers who will utilise this data to study doctrinal texts.

2.2. Sinhala Diachronic Corpus

The *Sinhala Diachronic Corpus Version-1.0* (SiDiaC-v.1.0) was introduced by Jayatilleke and de Silva (2025a) and comprises 58,027 word tokens drawn from 46 literary works spanning the 5th to 20th century CE (426 CE to 1944 CE), sourced from the *National Library of Sri Lanka*. Text extraction was carried out using Google Document AI OCR, achieving an average accuracy of 96.84% across all processed documents. The corpus was carefully constructed to ensure balanced temporal coverage whilst prioritising canonical literary works; among the 46 books included, 18 are religious texts predominantly focused on Buddhism, annotated with a two-layer genre scheme (Fiction/Non-Fiction at primary level; Religious, History, Poetry, Language, and Medical at secondary level). SiDiaC-v.1.0 implements copyright filtering based on Sri Lanka's 70-year post-mortem *auctoris* rule under the Intellectual Property Act No. 36 of 2003, a practice directly adopted by SiPaKosa.

The second version of this corpus, SiDiaC-v.2.0 (Jayatilleke et al., 2026), substantially expands the resource to 241k words across 185 literary works, with publication dates ranging from 1800 CE to 1955 CE and written dates spanning the 5th to the 20th century CE. A date-annotated subset of 59 documents totalling approximately 67k words is further stratified by estimated written date, enabling fine-grained diachronic analysis. Of the 185 documents, 86 are classified under the religious genre, reflecting the enduring centrality of Buddhism to Sinhala literary culture. SiDiaC-v.2.0 addresses several limitations of its predecessor, including more comprehensive post-processing to correct malformed tokens, code-mixed content (Pali, Sanskrit, and English), and multi-column rendering errors introduced by OCR; it also broadens the written-date annotation methodology from strict manuscript-composition dating to the publication-year approach used in corpora such as COHA (Davies, 2012), enabling inclusion of a wider range of literary works.

SiDiaC focuses on diachronic language change across secular literary traditions and provides the most directly comparable Sinhala corpus to SiPaKosa in terms of construction methodology.

2.3. Religious and Cultural Text Corpora

The digitisation of religious texts in low-resource languages presents unique challenges that require not only linguistic accuracy but also cultural sensitivity and the preservation of specialised terminology. Several major religious corpora initiatives have established important precedents across traditions.

The Digital Pali Canon provides comprehensive coverage of Theravāda scriptures in Pali,[3] whilst the *BDK English Tripiṭaka*[4] offers scholarly English translations of the Chinese Tripiṭaka. The *Chinese Buddhist Electronic Text Association* (CBETA) represents one of the most comprehensive Buddhist digitisation efforts globally, providing over 100 million characters of Chinese Buddhist texts drawn from the Taishō Shinshū Daizōkyō (Volumes 1-55 and 85; 5,320+ individual texts) and supplementary collections (Wittern, 2002). Files are encoded in TEI∼P5 XML with Unicode handling for 2,800+ rare Buddhist characters proposed to the Unicode Consortium, released under CC BY-NC-SA 2.5∼TW.

The GRETIL (*Göttingen Register of Electronic Texts in Indian Languages*) project provides over 1,000 electronic editions primarily in Sanskrit, with additional texts in Pali, Prakrit, Tamil, Malayalam, Hindi, Marathi, Old Javanese, and Tibetan (Maas, 2020). The total collection spans several gigabytes in plain text, HTML, and TEI-conformant XML, released under CC∼BY via Zenodo. However, GRETIL provides minimal annotation (raw text only) and an ongoing migration from legacy encodings; whilst it includes some Pali texts, it provides no NLP-ready annotation layers, unlike SiPaKosa.

The *Quranic Arabic Corpus* (Dukes and Habash, 2010) provides morphological annotation of all 77,430 words of the Qur'an across 128,219 morphemic forms (clitic-level segmentation) using a

[3] https://tripitaka.online/

[4] https://www.bdkamerica.org/tripitaka-list/

Corpus	Language(s)	Tradition	Tokens	Units	Annotation	Alignment
Bible 100 (2015)	100 langs	Christian	varies	∼31K verses/lang	None	Verse
CBETA (2002)	Classical Chinese	Buddhist	100M+ chars	5,320+ texts	TEI XML	-
GRETIL (2020)	10+ Indic	Multi	GBs (raw)	1,000+ texts	Raw text	-
Quranic Arabic (2010)	Arabic	Islamic	77,430 words	6,236 āyahs	Morph+Syn+Sem	Verse
Tanzil (2007)	40+ langs	Islamic	varies	6,236 verses/lang	None	Verse
DCS (2020)	Sanskrit	Multi	2.5M+ items	∼650K sents	Lemma+POS+Morph	-
SansTib (2022)	Skt-Tibetan	Buddhist	14.4M tokens	317K pairs	Aligned	Sentence
MITRA (2026)	Skt+Zh+Tib+Pali	Buddhist	varies	1.74M pairs	MT-aligned	Sentence
SiDiaC-v.2.0 (2026)	Sinhala	Mixed	241K words	185 works	Genre+date	-
SiPaKosa [our work]	Sinhala + Pali	Buddhist	**9.25M tokens**	**786K sents**	Lang. class + meta	-

Table 1: Comparison of SiPaKosa with related religious and Indic text corpora.

44-tag POS tagset derived from traditional Arabic grammar, spanning 114 surahs and 6,236 āyahs. The corpus provides four annotation layers: morphological segmentation, part-of-speech tagging, syntactic dependency analysis, and semantic ontology, making it the most methodologically comparable sacred-text annotation project to what SiPaKosa could aspire to in future extensions. Building upon this foundation, the *Quranic Arabic Dependency Treebank* (QADT) (Dukes et al., 2010) extends morphological analysis with gold-standard syntactic annotation for approximately 37,578 words (∼49% of the Qur'an), achieving an F-measure of 78% with a rule-based dependency parser trained on the annotated data.

The *Tanzil parallel corpus* (Tanzil Project, 2007) offers Qur'anic translations in over 40 languages with verse-level alignment across 6,236 verses and 114 surahs, facilitating cross-lingual Islamic studies. The *Hebrew Bible Treebank* (Sade et al., 2018) offers morphological and syntactic annotation of Biblical Hebrew with detailed morphological features following traditional Hebrew grammar. The *Digital Corpus of Sanskrit* (DCS) (Hellwig et al., 2020) contains approximately 650,000 sentences with over 2.5 million lexical items annotated for lemmatisation, POS tagging, and morphological analysis, covering Sanskrit texts from 500 BCE to 1900 CE; later extensions added syntactic annotation in Universal Dependencies format (Hellwig and Nehrdich, 2018) and word sense annotation. At 650k sentences, the DCS is the most directly comparable single-language annotated ancient-text corpus to SiPaKosa's 786k sentences, and both address low-resource ancient languages; however, DCS does not cover Buddhist Pali in the Sinhala script, nor does it address language mixing between a living vernacular and a classical liturgical language.

These projects collectively demonstrate shared requirements: (1) domain expertise for doctrinal accuracy, (2) preservation of specialised religious terminology, (3) careful handling of language mixing between sacred languages (e.g. Pali-Sinhala, Arabic-Urdu, Sanskrit-Hindi), and (4) maintenance of cultural context. Table 1 presents a systematic comparison of SiPaKosa against related religious and Indic text corpora across six dimensions: language coverage, religious tradition, scale, annotation depth, and alignment granularity.

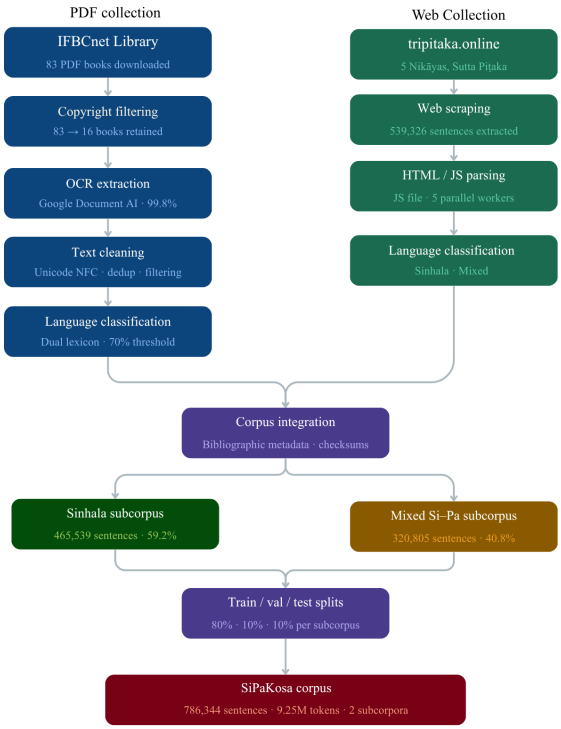


Figure 1: Complete methodology pipeline showing dual-source data collection, processing, and integration.

3. Methodology

We employ a dual-source methodology in order to cover canonical Buddhist scriptures with historically archived texts to achieve comprehensive coverage of Sinhala and Pali Buddhist literature. Our complete pipeline from data source identification through final corpus creation is illustrated in Figure 1. Furthermore, the book level filtering from a total of 83 to 16 eligible books is depicted in Figure 2.

3.1. Data Collection

Our historical corpus is compiled from the IFBCnet Library[5], a comprehensive digital archive of Sinhala and Pali Buddhist texts along with web-scrapping tripitaka.online[6], to complement historical texts with canonical scriptures. We downloaded 83 PDF books from IFBCnet Library, representing the complete available collection, each accompanied by bibliographic metadata including title, author information, publication year, and category classifications. The archive spans publications from the early days through the mid-20th century, providing valuable historical coverage of Sinhala and Pali Buddhist scholarship during a critical period of Buddhist revival and modernisation in Sri Lanka.

To ensure legal compliance, we implemented rigorous copyright filtering based on Sri Lanka's 70-year post-mortem auctoris rule under the Intellectual Property Act No. 36 of 2003[7] following the methodology by Jayatilleke et al. (2026). Firstly, Tripiṭaka canonical texts in PDF format were excluded via hardcoded rules, as these would be acquired separately through web scraping to ensure canonical completeness. Secondly, books authored by individuals who passed away before 1954 were classified as public domain with high confidence. Finally, books with living authors or incomplete author metadata were conservatively excluded to minimise legal risk. This conservative approach prioritises legal certainty over corpus size.

After copyright filtering, 16 books (19.3% of downloaded collection) were deemed suitable for inclusion. These books span three traditional Sinhala and Pali Buddhist literary categories: "Books related to Tipitaka" comprising 5 books (31.25%), which include canonical translations and commentaries; "Old Buddhist books" comprising 8 books (50.0%), which encompass philosophical treatises, devotional literature, and scholarly works; and "Buddhist characters" comprising 3 books (18.75%), which consist of hagiographies and biographical accounts of prominent Buddhist figures. The entire filtering process, from initial collection to the selection of books for the final corpus, is illustrated in Figure 2.

In addition to the historical PDF corpus, we systematically web-scraped tripitaka.online, a comprehensive digital repository of the Theravada Pali Canon maintained by Buddhist scholars, which provides Sinhala-language translations of the canonical scriptures alongside the original Pali

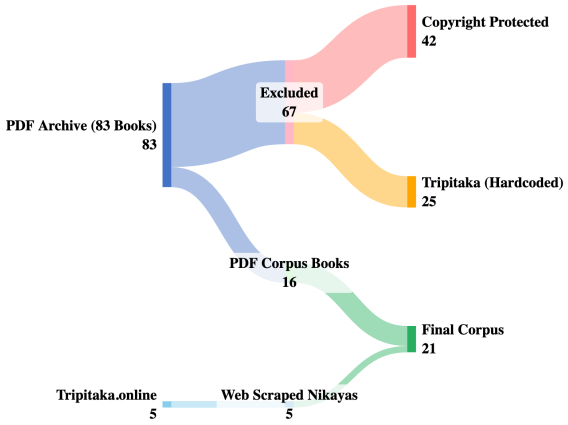


Figure 2: Book-level filtering from 83 downloaded books to 16 corpus-eligible books.

text. The Canon, known as the Tripiṭaka, literally *three baskets* in Pali, comprises the Vinaya Piṭaka (monastic discipline), the Sutta Piṭaka (discourses of the Buddha), and the Abhidhamma Piṭaka (systematic philosophical analysis). Our scraping focused on the five Nikayas of the Sutta Pitaka: Dīgha Nikāya (Long Discourses, 34 suttas), Majjhima Nikāya (Middle-Length Discourses, 152 suttas), Saṃyutta Nikāya (Connected Discourses, 65 suttas), Aṅguttara Nikāya (Numerical Discourses, 1,365 suttas), and Khuddaka Nikāya (Minor Collection, 831 suttas). Each sutta on tripitaka.online presents Pali canonical verses alongside corresponding Sinhala translations, providing natural parallel text alignment that preserves the semantic correspondence between source and target languages (this corpus lacks clear parallel text alignment but contains useful information for translation work). The detailed implementation is provided in Appendix B.1.

3.2. Text Extraction and Processing

We developed a two-stage extraction pipeline combining traditional PDF text extraction with OCR-based processing to handle the diverse quality of historical scans. For each PDF, we first attempted direct text extraction using pdfplumber, a Python library optimised for extracting text from PDF documents. As it did not produce precise extractions, we employed Google Document AI[8] (Jayatilleke and de Silva, 2025b), a production-grade OCR service with demonstrated strong performance on Indic scripts, including Sinhala (Jayatilleke and de Silva, 2025a).

This way, we ensure that if we can collect data from the PDF, we can get it without major issues or inconsistencies, but if the page is a scanned

[5] https://download.ifbcnet.org/

[6] https://tripitaka.online/

[7] https://www.gov.lk/wordpress/wp-content/uploads/2015/03/IntellectualPropertyActNo.36of2003Sectionsr.pdf

[8] https://cloud.google.com/document-ai/

image, we have to use the alternative route of OCR extraction.

For pages processed through Document AI, we implemented comprehensive quality control measures based on character-level confidence scores returned by the OCR engine. The OCR process achieved an average character-level confidence of 99.8% across all processed pages, demonstrating both the quality of the source scans and Document AI's effectiveness on Sinhala script. Pages were classified into two confidence tiers: high confidence (mean confidence >=0.85, representing 85% of OCR-processed pages) and low confidence (mean confidence < 0.85, representing 15% of pages). Low-confidence pages were flagged for potential manual review and excluded from the primary corpus but preserved for future improvement efforts.

The proportion of pages achieving high confidence (85%) demonstrates both the quality of the archive scans and Document AI's effectiveness on Sinhala script. We also implemented automated page classification to identify and filter non-content pages, such as covers, title pages, tables of contents, indices, and blank pages, using pattern-matching rules, retaining 7,064 content pages (99.8% retention).

For web-scraped canonical texts, we implemented a systematic crawler to extract structured HTML content from tripitaka.online, whilst preserving both Pali canonical text and Sinhala translations separately. After careful analysis, we identified that the data displayed on tripitaka.online, comes from a js (JavaScript) file, so we made sure the web scraper algorithm took this js file and read its content to get the textual data we needed, we preserved structural metadata during this process to organise the whole process which included nikaya identifiers, book numbers, sutta identifiers, and verse numbers for precise alignment and reference purposes. The extraction process required 72 hours, using multithreading with 5 parallel workers and rate-limiting constraints to ensure server stability, reflecting the substantial scale of the canonical corpus (539,326 sentences across five Nikayas).

3.3. Language Classification

Buddhist texts in Sri Lanka traditionally mix Sinhala and Pali with varying proportions depending on text type, historical period, and intended audience. To facilitate both monolingual and cross-lingual research, we implemented an automated language classification mechanism using lexicon-based matching. We constructed two primary lexicons: a Sinhala lexicon based on *Google's open-source Sinhala Pronunciation Lexicon*[9] containing 41,617 word forms covering a vast vocabulary, and a custom Pali lexicon containing 14,278 terms extracted from a Sinhala-Pali dictionary, which was included as one of the 16 books and then filtered for uniqueness to avoid overlap with Sinhala words. Further information on constructing the custom Pali Lexicon can be found at Appendix B.3

Pages and sentences were classified using a combined confidence scoring system that integrates lexicon-based word matching with morphological pattern analysis. For each text segment, we calculated a weighted confidence score for both Sinhala and Pali, where lexicon coverage (proportion of words matching each language's lexicon) received 70% weight and morphological feature detection (case markers, verbal endings, particles) received 30% weight. Text was classified as Sinhala only if the Sinhala confidence score reached at least 70% and exceeded the Pali confidence score. Text was classified as Pali if the Pali confidence score reached at least 70% and exceeded the Sinhala confidence score. Text was classified as Mixed if neither language's confidence score reached the 70% threshold, indicating substantial presence of both languages or ambiguous linguistic features. The 70% threshold was selected empirically after manual inspection of sample texts to balance precision (avoiding false classifications) and recall (capturing genuine monolingual content).

3.4. Corpus Integration and Metadata

We combined sentences from both sources into unified language-based subcorpora designed to support diverse research applications. The Sinhala subcorpus combines historical PDF Sinhala pages (111,632 sentences) with Tripitaka Sinhala translations (353,907 sentences), totalling 465,539 sentences representing purely Sinhala content suitable for monolingual language modelling and Sinhala-specific NLP tasks. The Mixed subcorpus combines historical PDF mixed pages (135,386 sentences) with Tripitaka Pali canonical text (185,419 sentences), totalling 320,805 sentences in both Sinhala and Pali that capture the language-mixing characteristic of Buddhist scholarly discourse.

For each book in the corpus, we collected comprehensive metadata in accordance with best practices in digital humanities corpus construction (Jayatilleke and de Silva, 2025a). The schema captures five categories of information: bibliographic details (source URLs, Sinhala and English titles, publication dates, and publisher information); author records (full names, dates of birth and death, and biographical notes); technical provenance (file

[9] https://raw.githubusercontent.com/google/language-resources/master/si/data/lexicon.tsv

sizes, SHA-256 checksums, download timestamps, and OCR confidence statistics); classification labels (category assignments, language classification results, and historical period); and copyright status (public domain eligibility and confidence level of assessment). The full metadata schema is provided in Figure 5.

We created train-validation-test splits at 80-10-10 ratios for each language subcorpus. For the Sinhala subcorpus, this resulted in 372,431 sentences for training, 46,553 sentences for validation, and 46,555 sentences for testing. For the Mixed subcorpus, this yielded 256,644 for training, 32,080 sentences for validation, and 32,081 sentences for testing. The training set comprises 629,075 sentences (80% of the total corpus), the validation set comprises 78,633 sentences (10%), and the test set comprises 78,636 sentences (10%) when aggregated across all language subcorpora and converted to sentence-level statistics. These splits are fixed and are released with the corpus to enable direct comparison of future research results.

4. Evaluation of SiPaKosa

4.1. Corpus Statistics

SiPaKosa comprises 786,839 total sentences from dual complementary sources: 247,513 sentences extracted from 16 copyright-cleared historical PDF books spanning 7,064 content pages, and 539,326 sentences from five Nikayas of web-scraped canonical texts. A comprehensive statistical analysis of the complete integrated corpus is presented in Table 2.

Source	Metric	Count
Historical PDF Corpus	Documents (books)	16
	Content pages	7,064
	Sentences	247,513
Web-Scraped Canonical Texts	Nikayas covered	5
	Sentences	539,326
Total	**Sentences**	**786,344**
	Tokens	**9,249,792**
	Average tokens/sentence	**11.76**

Table 2: Overall corpus statistics combining historical and canonical texts.

4.1.1. Language Distribution

The Sinhala subcorpus dominates at 59.2% of the total corpus (465,539 sentences), reflecting our focus on Sinhala Buddhist literature. The mixed subcorpus comprises 40.8% (320,805 sentences), capturing canonical Pali scriptures and mixed commentarial traditions. The statistics of each language subcorpus, revealing the complementary contributions of our dual sources, are detailed in Table 3.

The Sinhala subcorpus (465,539 total sentences) is predominantly web-scraped (76% from Tripitaka, 24% from PDF), reflecting the extensive online availability of canonical translations. The Mixed subcorpus (320,805 total) shows more balanced contributions (58% web-scraped, 42% PDF), capturing both canonical Pali verses and historical commentarial discourse. Based on our methodology for language classification in 3.3, there was a possibility of 3 language classes forming; unfortunately, no Pali-only content was found. The dual-source composition of each language subcorpus is illustrated in Figure 3.

Type	Metric	Sinhala	Mixed
Sentences	PDF corpus	111,632	135,386
	Web-scraped	353,907	185,419
	Total Count	**465,539**	**320,805**
	% of corpus	59.2%	40.8%
Tokens	**Total Count**	**5,418,716**	**3,831,076**
	% of corpus	58.6%	41.4%

Table 3: Sentence and token distribution by language subcorpus and source.

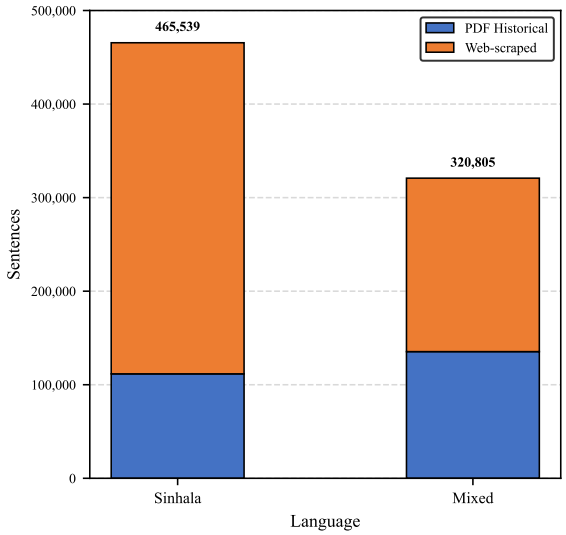


Figure 3: Sentence distribution by language and source.

At the token level, the corpus comprises 9,253,009 tokens, with the Sinhala subcorpus containing 5,418,716 tokens (58.6%) and the Mixed subcorpus containing 3,831,076 tokens (41.4%). The average sentence length varies across subcorpora: 11.6 tokens per sentence for Sinhala, 11.9 tokens per sentence for Mixed text.

4.2. Results

4.2.1. Evaluation Setup

We evaluate corpus quality and domain characteristics using nine state-of-the-art language models selected to represent diverse architectures, param-

Model	Buddhist Sinhala	Mixed Si-Pa	General Sinhala	BS/G Ratio	M/G Ratio
GPT-3.5-Turbo	**1.09**	**1.11**	**1.08**	1.01	1.03
GPT-4o-mini	1.23	1.15	1.31	0.94	0.88
GPT-4-Turbo	1.56	1.59	1.48	1.05	1.07
GPT-4o	1.62	1.77	1.68	0.96	1.06
Llama-3.1-8B-Instruct	3.29	4.18	3.26	1.01	1.28
Aya-Expanse-8B	6.21	8.82	4.90	1.27	1.80
Llama-3.2-3B-Instruct	6.62	7.81	6.94	0.95	1.13
Qwen2.5-3B-Instruct	7.24	9.81	6.56	1.10	1.49
Gemma-2-9B-It	22.21	36.71	14.85	1.50	2.47
SinLlama	183.33	189.67	101.57	1.80	1.87

Table 4: Comparison of perplexity among proprietary language models and small language models using different corpora.

eter scales, and training paradigms. Proprietary models include models from the GPT family; GPT-3.5-Turbo, GPT-4o-mini, GPT-4-Turbo, and GPT-4o (OpenAI, 2023). Open-source models include Llama-3.1-8B-Instruct and Llama-3.2-3B-Instruct (Grattafiori et al., 2024), aya-expanse-8b (Dang et al., 2024), Qwen2.5-3B-Instruct (Hui et al., 2024), and Gemma-2-9B-it (Team et al., 2024) with explicit multilingual and Sinhala support.

We evaluate models on three carefully constructed test sets designed to assess different aspects of Sinhala language modelling capability. The Buddhist Sinhala test set comprises 1,024 sentences sampled using diversity sampling from the pure Sinhala test split. The Mixed Sinhala-Pali test set comprises 1,024 sentences from the mixed test split, representing mixed Buddhist scholarly discourse. The General Sinhala test set comprises 1,024 sentences from a CulturaX[10], which consists of Sinhala news articles and online content. To ensure representative coverage while avoiding redundancy, we employ diversity sampling using K-means clustering (k=4) on multilingual sentence embeddings, selecting sentences closest to cluster centroids to capture diverse linguistic phenomena within each corpus.

We report corpus-level perplexity calculated as the exponentiated average negative log-likelihood across all tokens in each test set. For open-source models with direct access to model internals, given a test set with N tokens $w_1, w_2, \ldots, w_N$, perplexity is computed as;

$$\mathbf{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i|w_{<i})\right) \quad (1)$$

where $P(w_i|w_{<i})$ is the model's predicted probability for token w_i given preceding context. For proprietary models that do not expose direct token-level log probabilities, we employ an approximate evaluation methodology in which models assess linguistic quality per sentence, with response confidence serving as a proxy for model certainty about the input text. Detailed proprietary model evaluation methodology is provided in Appendix C.1. To quantify domain-specific challenges independent of absolute perplexity values, we compute domain gap ratios as;

$$\mathbf{BS/G} = \frac{\mathrm{PPL}_{\mathrm{Buddhist-Sinhala}}}{\mathrm{PPL}_{\mathrm{General}}} \quad (2)$$

$$\mathbf{M/G} = \frac{\mathrm{PPL}_{\mathrm{Mixed}}}{\mathrm{PPL}_{\mathrm{General}}} \quad (3)$$

A comprehensive perplexity results across all nine models and three test corpora are presented in Table 4. The results reveal substantial performance variation between proprietary models and open-source alternatives.

4.2.2. Model Performance Analysis

GPT-3.5-Turbo achieves the best overall performance with the lowest perplexities of 1.09 for Buddhist Sinhala, 1.11 for Mixed Sinhala-Pali, and 1.08 for General Sinhala, demonstrating strong Sinhala language comprehension. All four proprietary models substantially outperform all locally-hosted open-source models, with GPT-3.5-Turbo, GPT-4o-mini, GPT-4-Turbo, and GPT-4o achieving perplexities between 1.08 and 1.77 across all test sets.

Among open-source models, Llama-3.2-3B-Instruct leads with perplexities of 3.29 for Buddhist and 4.18 for Mixed corpora, representing the best available option for resource-constrained settings. Aya-Expanse-8B, despite explicit multilingual training including Sinhala, achieves moderate performance (6.21 Buddhist, 8.82 Mixed), whilst

[10] https://huggingface.co/datasets/uonlp/CulturaX

smaller 3B models (`Llama-3.2-3B-Instruct`, `Qwen2.5-3B-Instruct`) demonstrate competent performance. `Gemma-2-9B-It` significantly underperforms with perplexities exceeding 14 on all corpora despite its 9B parameter count, indicating fundamentally inadequate Sinhala coverage in pretraining data.

Domain gap ratios reveal how models handle the specialised vocabulary and discourse patterns of Buddhist literature compared to general Sinhala. `GPT-3.5-Turbo` exhibits minimal domain gap with a BS/G ratio of 1.01 and M/G ratio of 1.03, suggesting a promising generalisation across registers and domains. `GPT-4o-mini` exhibits an inverse domain gap on Buddhist texts (BS/G ratio of 0.94), actually performing better on classical Buddhist Sinhala than general text. `Llama-3.1-8B` shows generalisation on pure Sinhala (BS/G ratio 1.01) but moderate difficulty with mixed text (M/G ratio 1.28). `Aya-Expanse-8B` and `Gemma-2-9B` exhibit larger domain gaps (1.27-2.47), with Gemma's extreme M/G ratio of 2.47 indicating the comprehension difficulty with language mixing.

Mixed Sinhala-Pali corpora present substantially elevated perplexity across most models, with an average M/G ratio of 1.42 across all models excluding the `Gemma` outlier. This consistent difficulty possibly reflects fundamental challenges in handling intra-sentential language mixing, in which Pali canonical quotations and Sinhala explanatory prose alternate rapidly. Proprietary models handle language mixing most effectively with M/G ratios ranging from 0.88 to 1.07. Open-source models struggle more significantly: `Llama-3.1` shows 28% perplexity increase (M/G ratio 1.28), `Aya-Expanse` shows 80% increase (1.80), and `Qwen2.5` shows 49% increase (1.49).

Another finding of this evaluation is the substantial performance gap between proprietary models and locally hosted open-source alternatives. `GPT-3.5-Turbo` outperforms the best local model `Llama-3.1-8B` by a factor of 3.0 on Buddhist Sinhala (perplexity 1.09 vs 3.29). This gap persists across all three test sets, suggesting it reflects fundamental differences in the scale and quality of the training data. The gap has critical practical implications: whilst proprietary models offer exceptional performance for production Buddhist NLP applications, they incur per-token costs and lack transparency. Open-source alternatives enable offline deployment, fine-tuning for specific Buddhist NLP tasks, and cost-effective experimentation.

Our findings for open source models challenge the common belief that larger models always outperform smaller ones, showing more complex relationships between parameter count and performance in low-resource languages. `Llama-3.2-3B` (perplexity 6.62) outperforms the larger `Aya-Expanse-8B` (6.21) despite having fewer than half the parameters, whilst `Gemma-2-9B`'s catastrophic performance (22.21) demonstrates that scale alone provides no guarantees. For proprietary models where accuracy is paramount, and API costs are acceptable, `GPT-3.5-Turbo` and `GPT-4o-mini` provide powerful performance. For research systems requiring local deployment and customisation, `Llama-3.1-8B` represents the best foundation despite its 3 times higher perplexity. In resource-constrained settings, `Llama-3.2-3B` or `Qwen2.5-3B` offer strong performance with manageable computational requirements.

5. Conclusion

We have presented SiPaKosa, a comprehensive corpus of canonical Sinhala and Pali Buddhist texts compiled from dual sources: historical public domain archives and systematically web-scraped canonical scriptures. Through rigorous digitisation, OCR processing, web scraping, copyright filtering, and quality assurance, we created a high-quality resource comprising 786,839 sentences (9.25 million tokens) across Sinhala, Pali, and mixed subcorpora, addressing critical gaps in Sinhala NLP while preserving important cultural and religious heritage.

Our comprehensive baseline evaluations across nine language models demonstrate significant performance variation, with proprietary models achieving remarkably low perplexities of 1.08 to 1.77 and substantially outperforming open-source alternatives with perplexities of 3.29 to 36.71. This performance gap highlights both the current state of Sinhala language modelling and opportunities for improvement in open-source multilingual systems. All models struggle with Sinhala-Pali mixed text, indicating opportunities for specialised model development targeting classical Buddhist literature.

This corpus serves as a critical resource for diachronic linguistic studies, providing insights into the history and evolution of language, as well as the profound influence of religious thought on everyday speech. In addition to formal canonical analysis, this corpus allows examination of folk speech and literary proverbs derived from classical Buddhist texts (Sofalas et al., 2026). This reflects the historical process by which scriptural concepts were integrated into the broader Sinhala linguistic identity.

Limitations

Temporal and Genre Coverage: Copyright restrictions exclude scholarly works published after the 1950s, potentially missing modern interpretations and contemporary Buddhist scholarship.

Post-Processing: Although the OCR extraction pipeline achieved an average character-level confidence of 99.8% across all processed pages, no manual post-processing was conducted on the extracted text. Correcting residual OCR errors and normalising historical spelling variations in classical Sinhala requires deep linguistic expertise in diachronic Sinhala and Pāli that was not available to the authors; consequently, such errors and orthographic inconsistencies may persist in the corpus.

Language Classification: Buddhist texts frequently mix Sinhala and Pali within sentences, making strict language separation imperfect. The Mixed category encompasses diverse text types, from balanced bilingual to predominantly Pali canonical texts. The 70% classification threshold is somewhat arbitrary and may benefit from optimisation. Additionally, this study did not explore potential Sanskrit text in these documents due to the absence of advanced language classification models.

Evaluation Limitations: Standard perplexity does not directly measure downstream task performance or semantic understanding. We lack human assessments of model outputs on Buddhist question answering or text generation tasks. Cannot analyse internal mechanisms of proprietary models or verify their Sinhala training data. Could evaluate other proprietary models rather than only OpenAI models

6. Bibliographical References

David Ifeoluwa Adelani, Jessica Ojo, Israel Abebe Azime, Jian Yun Zhuang, Jesujoba Oluwadara Alabi, Xuanli He, Millicent Ochieng, Sara Hooker, Andiswa Bukula, En-Shiun Annie Lee, Chiamaka Ijeoma Chukwuneke, Happy Buzaaba, Blessing Kudzaishe Sibanda, Godson Koffi Kalipe, Jonathan Mukiibi, Salomon Kabongo Kabenamualu, Foutse Yuehgoh, Mmasibidi Setaka, Lolwethu Ndolela, Nkiruka Odu, Rooweither Mabuya, Salomey Osei, Shamsuddeen Hassan Muhammad, Sokhar Samb, Tadesse Kebede Guge, Tombekai Vangoni Sherman, and Pontus Stenetorp. 2025. IrokoBench: A new benchmark for African languages in the age of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2732–2757, Albuquerque, New Mexico. Association for Computational Linguistics.

Muhammad Farid Adilazuarda, Sagnik Mukherjee, Pradhyumna Lavania, Siddhant Shivdutt Singh, Alham Fikri Aji, Jacki O'Neill, Ashutosh Modi, and Monojit Choudhury. 2024. Towards measuring and modeling "culture" in LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15763–15784, Miami, Florida, USA. Association for Computational Linguistics.

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2024. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 12:681–699.

HWK Aravinda, Rashad Sirajudeen, Samith Karunathilake, Nisansa de Silva, Rishemjit Kaur, and Surangika Ranathunga. 2025. Sinllama-a large language model for sinhala. In *2025 Moratuwa Engineering Research Conference (MERCon)*, pages 617–622. IEEE.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23, Doha, Qatar. Association for Computational Linguistics.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Andreas Chari, Sean MacAvaney, and Iadh Ounis. 2025. Improving low-resource retrieval effectiveness using zero-shot linguistic similarity transfer. In *European Conference on Information Retrieval*, pages 290–306. Springer.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, et al. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

Mark Davies. 2012. Expanding horizons in historical linguistics with the 400-million word corpus of historical american english. *Corpora*, 7(2):121–157.

Nisansa De Silva. 2026. Survey on publicly available sinhala natural language processing tools and research. *arXiv preprint arXiv:1906.02358*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Vinura Dhananjaya, Piyumal Demotte, Surangika Ranathunga, and Sanath Jayasena. 2022. BERTifying Sinhala - a comprehensive analysis of pre-trained language models for Sinhala text classification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7377–7385, Marseille, France. European Language Resources Association.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada. Association for Computational Linguistics.

Kais Dukes, Eric Atwell, and Abdul-Baquee M. Sharaf. 2010. Syntactic annotation guidelines for the Quranic Arabic dependency treebank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Kais Dukes and Nizar Habash. 2010. Morphological annotation of Quranic Arabic. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier L de Lacalle, and Mikel Artetxe. 2024. Bertaqa: How much do language models know about local culture? *Advances in Neural Information Processing Systems*, 37:34077–34097.

James W Gair. 1996. Sinhala writing. *The world's writing systems*, pages 408–412.

Iker García-Ferrero, Rodrigo Agerri, and German Rigau. 2025. Cross-lingual transfer for low-resource natural language processing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Viktor Hangya and Alexander Fraser. 2022. Improving low-resource languages in pre-trained multilingual language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 2872–2885.

Oliver Hellwig and Sebastian Nehrdich. 2018. Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2754–2763, Brussels, Belgium. Association for Computational Linguistics.

Oliver Hellwig, Salvatore Scarlata, Elia Ackermann, and Paul Widmer. 2020. The treebank of vedic Sanskrit. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5137–5146, Marseille, France. European Language Resources Association.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Zhiqi Huang, Hansi Hettiarachchi Ng, and Hamed Chen. 2023. Improving cross-lingual information retrieval on low-resource languages via optimal transport distillation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4561–4573.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Ben Hutchinson. 2024. Modeling the sacred: Considerations when using religious texts in natural language processing. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1029–1043, Mexico City, Mexico. Association for Computational Linguistics.

Nevidu Jayatilleke and Nisansa de Silva. 2025a. SiDiaC: Sinhala diachronic corpus. In *Proceedings of the 39th Pacific Asia Conference on Language, Information and Computation*, pages 511–527, Hanoi, Vietnam. Association for Computational Linguistics.

Nevidu Jayatilleke and Nisansa de Silva. 2025b. Zero-shot OCR accuracy of low-resourced languages: A comparative analysis on Sinhala and Tamil. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era*, pages 471–480, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Nevidu Jayatilleke, Nisansa de Silva, Uthpala Nimanthi, Gagani Kulathilaka, Azra Safrullah, and Johan Sofalas. 2026. SiDiaC-v.2.0: Sinhala diachronic corpus version 2.0. *arXiv preprint arXiv:2603.10861*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Jürgen Knauth and David Alfter. 2014. A dictionary data processing environment and its application in algorithmic processing of Pali dictionary data for future NLP tasks. In *Proceedings of the Fifth Workshop on South and Southeast Asian Natural Language Processing*, pages 65–73, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. ArabicMMLU: Assessing massive multitask language understanding in Arabic. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5622–5640, Bangkok, Thailand. Association for Computational Linguistics.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M Khapra, Pratyush Kumar, et al. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*.

Dongyub Lee, Younghun Jeong, Hwa-Yeon Kim, Hongyeon Yu, Seunghyun Han, Taesun Whang, Seungwoo Cho, Chanhee Lee, Gunsu Lee, and Youngbum Kim. 2024. Tree-of-question: Structured retrieval framework for Korean question answering systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 406–418, Mexico City, Mexico. Association for Computational Linguistics.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, Bangkok, Thailand. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Philipp André Maas. 2020. Pātañjalayogaśāstram. göttingen register of electronic texts in indian languages (gretil).

Sebastian Nehrdich. 2022. SansTib, a Sanskrit - Tibetan parallel corpus and bilingual sentence embedding model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6728–6734, Marseille, France. European Language Resources Association.

Sebastian Nehrdich and Kurt Keutzer. 2026. Mitra: A large-scale parallel corpus and multilingual pretrained language model for machine translation and semantic retrieval for p\= ali, sanskrit, buddhist chinese, and tibetan. *arXiv preprint arXiv:2601.06400*.

OpenAI. 2023. Gpt-4 technical report. Technical report, OpenAI.

Soon Chang Poh, Sze Jue Yang, Jeraelyn Ming Li Tan, Lawrence Leroy Tze Yao Chieng, Jia Xuan Tan, Zhenyu Yu, Foong Chee Mun, and Chee Seng Chan. 2024. MalayMMLU: A multitask benchmark for the low-resource Malay language. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 650–669, Miami, Florida, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+

questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan Ak, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Surangika Ranathunga and Nisansa de Silva. 2022. Some languages are more equal than others: Probing deeper into the linguistic disparity in the NLP world. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 823–848, Online only. Association for Computational Linguistics.

François Rémy, Julia Platzer, and Marco Dinarelli. 2024. Transtokenization and cross-lingual vocabulary transfers: Language adaptation of llms for low-resource nlp. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3456–3471.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sacred Texts Archive. 2020. The Internet Sacred Text Archive. Available at `https://www.sacred-texts.com`.

Shoval Sade, Amit Seker, and Reut Tsarfaty. 2018. The Hebrew Universal Dependency treebank: Past present and future. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Semantic Bible Project. 2015. The Semantic Bible: A multilingual semantic role annotation. Available at `http://semanticbible.com`.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. 2025. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Johan Nevin Sofalas, Dilushri Pavithra, Nevidu Jayatilleke, and Ruvan Weerasinghe. 2026. SinFoS: A parallel dataset for translating Sinhala figures of speech. In *Proceedings of the 22nd Workshop on Multiword Expressions (MWE 2026)*, pages 8–26, Rabat, Marocco. Association for Computational Linguistics.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2025. KMMLU: Measuring massive multitask language understanding in Korean. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104, Albuquerque, New Mexico. Association for Computational Linguistics.

Tanzil Project. 2007. Tanzil quran text (version 1.1).

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Nārada Thera and B Veerabhadra Rao. 1954. *The Dhammapada*. John Murray London.

Sshubam Verma, Mohammed Safi Ur Rahman Khan, Vishwajeet Kumar, Rudra Murthy, and Jaydeep Sen. 2025a. MILU: A multi-task Indic language understanding benchmark. pages 10076–10132.

Sshubam Verma, Mohammed Safi Ur Rahman Khan, Vishwajeet Kumar, Rudra Murthy, and Jaydeep Sen. 2025b. MILU: A multi-task Indic language understanding benchmark. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10076–

10132, Albuquerque, New Mexico. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Christian Wittern. 2002. Chinese buddhist texts for the new millenium—the chinese buddhist electronic text association (cbeta) and its digital tripitaka.

Arda Yüksel, Abdullatif Köksal, Lütfi Kerem Senel, Anna Korhonen, and Hinrich Schuetze. 2024. TurkishMMLU: Measuring massive multitask language understanding in Turkish. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7035–7055, Miami, Florida, USA. Association for Computational Linguistics.

Wajdi Zaghouani, Abdelati Hawwari, and Mona Diab. 2012. A pilot PropBank annotation for Quranic Arabic. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 78–83, Montréal, Canada. Association for Computational Linguistics.

Dan Zigmond. 2023. Distinguishing commentary from canon: Experiments in pāli computational linguistics. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*, pages 213–222.

A. Additional Related Work

A.1. Multilingual Evaluation Benchmarks

Multilingual evaluation frameworks include XGLUE (Liang et al., 2020) (11 NLU and NLG tasks across 19 languages) and XTREME-R (Ruder et al., 2021) (10 tasks across 50 languages) for cross-lingual NLU tasks, building on English-only benchmarks GLUE (Wang et al., 2018) (9 NLU tasks, English only), SuperGLUE (Wang et al., 2019) (8 NLU tasks, English only), and SQuAD (Rajpurkar et al., 2016) (100,000+ extractive QA pairs, English only).

Language-specific MMLU variants have proliferated, covering Arabic (Koto et al., 2024), Chinese (Li et al., 2024), Turkish (Yüksel et al., 2024), Indonesian (Verma et al., 2025a), Korean (Son et al., 2025). Low-resource language benchmarks include AfriMMLU (Adelani et al., 2025) for African languages, MalayMMLU (Poh et al., 2024), BertaQA for Basque (Etxaniz et al., 2024), and MILU for 11 Indic languages (Verma et al., 2025b). The Include benchmark (Singh et al., 2025) attempts comprehensive multilingual coverage, though many benchmarks rely on automatic translation that can introduce errors and fail to capture cultural context (Adilazuarda et al., 2024).

Beyond the major religious corpus projects discussed in Sections 2.3-2.5, several specialised initiatives contribute to computational religious studies.

The Quranic PropBank (Zaghouani et al., 2012) extends semantic role labelling to Classical Arabic religious texts.

Christian Resources. The Semantic Bible project provides multilingual semantic role annotation for Biblical texts with ontological relationships (Semantic Bible Project, 2015).

Hindu and Vedic Resources. The Sansknet[11] project provides computational resources for Sanskrit including morphological analysers adapted to Vedic grammar.

Multi-Traditional Resources. The Sacred Texts Archive (Sacred Texts Archive, 2020) provides digitised versions of religious texts across 12 major traditions, though without linguistic annotation.

A.2. Indic Language Resources

SIPAKOSA exists within a vibrant Indic NLP ecosystem. Samanantar (Ramesh et al., 2022) is the largest publicly available Indic parallel corpus, comprising 49.7 million sentence pairs across 11 Indic languages; however, Pali is entirely absent from the collection, and the general-domain web-crawled text differs fundamentally from the specialised Buddhist canonical register that SIPAKOSA targets. IndicCorp v2 (Doddapaneni et al., 2023) provides 20.9 billion tokens across 24 Indic languages with the IndicXTREME benchmark (105 evaluation sets, 9 tasks); it includes Sinhala but not Pali, and its web-crawled sources contain no Buddhist domain content. The AI4Bharat-IndicNLP Corpus (Kunchukuttan et al., 2020) established 2.7 billion words for 10 Indic languages with pre-trained embeddings, providing foundational resources for the broader Indic NLP ecosystem.

These resources collectively provide broad coverage of general-domain Sinhala web text but offer zero coverage of Pali and zero domain-specific

[11] https://www.wilbourhall.org/sansknet/

Buddhist text, confirming that SiPaKosa addresses a genuine and unmet need within the Indic NLP landscape.

A.3. Language Identification

Since Sinhala and Pali share the same script (the Sinhala script is also used for writing Pali and Sanskrit in Sri Lanka; Gair 1996), language identification is a core technical challenge for SiPaKosa. Barman et al. (2014) established that approximately 7% of tokens in code-mixed text are ambiguous, a finding relevant to the Mixed Sinhala-Pali subcorpus.

A.4. Information Retrieval

Dense passage retrieval (Karpukhin et al., 2020) establishes neural retrieval baselines for open-domain question answering. For low-resource languages, optimal transport distillation (Huang et al., 2023) and zero-shot linguistic similarity transfer (Chari et al., 2025) improve cross-lingual IR effectiveness. Structured retrieval frameworks (Lee et al., 2024) provide QA methodologies that could be adapted for Sinhala Buddhist question answering.

A.5. Domain Adaptation

Parameter-efficient fine-tuning approaches, including LoRA (Hu et al., 2022) and QLoRA (Dettmers et al., 2023), enable domain adaptation without full fine-tuning, while knowledge distillation (Gou et al., 2021) supports creating smaller domain-specific models. These techniques are directly applicable to adapting general-purpose Sinhala models such as `SinBERT` (Dhananjaya et al., 2022) and `SinLlama` (Aravinda et al., 2025) for Buddhist domain tasks using SiPaKosa as the training corpus.

A.6. Cross-Lingual Transfer

García-Ferrero et al. (2025) provide comprehensive analysis of cross-lingual transfer techniques, whilst TransTokenization (Rémy et al., 2024) demonstrates vocabulary initialisation benefits for low-resource language adaptation of LLMs. Hangya and Fraser (2022) shows strategies for improving low-resource languages in multilingual models through vocabulary adaptation and continued pretraining, findings directly applicable to extending multilingual models for Sinhala Buddhist text.

A.7. LLM Evaluation

LLM evaluation surveys (Chang et al., 2024; Adlakha et al., 2024) discuss correctness and faithfulness assessment in instruction-following models, informing our baseline evaluation methodology. Hutchinson (2024) specifically addresses ethical and methodological considerations when using religious texts in NLP, including data provenance, cultural contexts, and proselytism concerns; this work directly informs SiPaKosa's commitment to copyright compliance and respectful representation of Buddhist canonical materials.

B. Additional Methodology

B.1. Web-scrapping

The web crawler for `tripitaka.online` implemented the following technical specifications: Rate limiting was set to 2 requests per second with exponential backoff retry logic (initial delay: 1s, maximum delay: 60s) to handle transient network failures. The crawler navigated the four-level hierarchy (Nikaya → Book → Sutta → Verse), extracted both Pali canonical text and Sinhala translations separately from the `JavaScript` file and preserved structural metadata, including nikaya identifiers, book numbers, sutta identifiers, and verse numbers for precise alignment. We employed a multi-threaded strategy with 5 parallel workers scraping 5 different suttas simultaneously, reducing total extraction time from an estimated 360 hours (sequential processing) to 72 hours (parallel processing), a 5× speedup. However, even with this optimisation, the extraction required 3 days of continuous operation due to the corpus scale (539,326 sentences) and necessary rate limiting (2 requests per second with exponential backoff) to avoid overwhelming the source server.

B.2. PDF Processing Flowchart

Figure 4 illustrates the end-to-end extraction pipeline applied to each of the 16 copyright-cleared PDF books. The pipeline begins by checking whether a book has already been processed, allowing interrupted runs to resume without reprocessing. Books that pass the 70-year post-mortem copyright check are then routed through one of two extraction paths depending on page content: digitally typed pages are handled by `pdfplumber`, while scanned pages are passed to `Google Document AI`, which returns character-level OCR confidence scores alongside the extracted text. All pages, regardless of extraction route, are classified into one of four structural categories: cover, table of contents, content, or index, and pages falling below the 0.7 confidence threshold are quarantined rather than discarded, preserving them for potential future review. Content pages that pass the confidence gate are written to disk as raw text, filtered to remove headers and OCR artefacts, and their provenance and quality metrics are recorded back into the per-book metadata record. The final output for

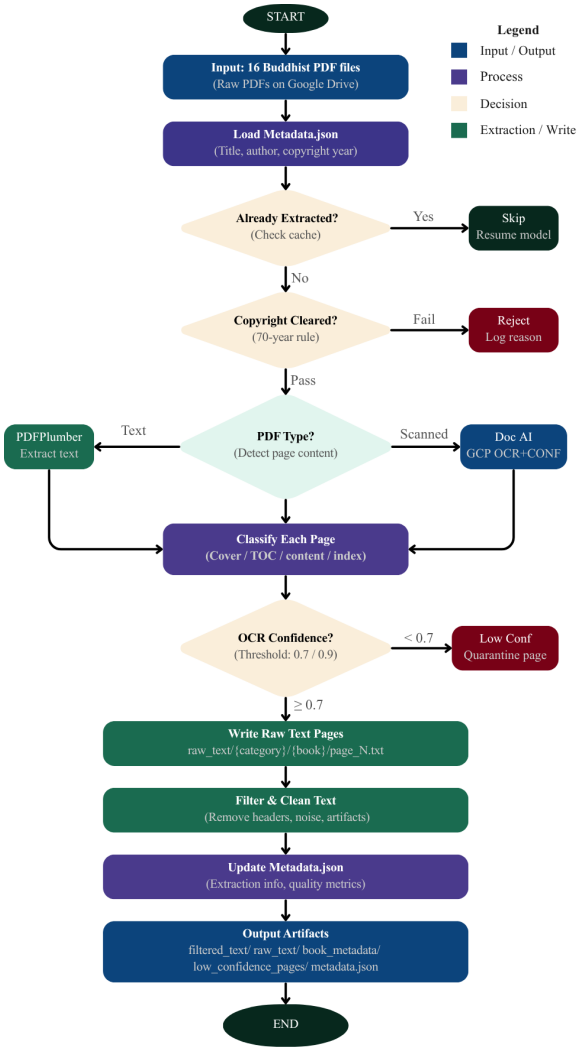


Figure 4: PDF processing pipeline applied to the 16 copyright-cleared historical Buddhist books, combining `pdfplumber` text extraction and Google Document AI OCR, with copyright checking, page classification, and confidence-based quality filtering (threshold: 0.7) prior to corpus integration.

each book comprises four artefact directories and an updated `metadata.json` file.

B.3. Custom Pali Lexicon

No publicly available Pali lexicon suitable for language classification existed at the time of this research, necessitating the construction of a custom resource. The source material was a Sinhala-Pali dictionary recovered as one of the 16 copyright-cleared books retained in the extraction pipeline described in Section 3.2 (Refer Table 5). All headword entries and their corresponding Pali equivalents were extracted from the dictionary's structured text output produced by the OCR stage, yielding a raw lexicon of 14,663 unique word forms. To isolate vocabulary that is diagnostically Pali rather than shared with modern Sinhala, the raw lexicon was filtered against a 41,617-word Sinhala lexicon obtained from Google's open-source language resources repository. Any token present in the Sinhala lexicon was removed from the Pali word list, on the basis that a word attested in the standard Sinhala vocabulary is not reliably diagnostic of Pali in a language classification context. The overlap analysis identified 887 shared words (6.05% of the raw Pali lexicon), predominantly short cognates of three to seven characters that are common to both languages due to their shared Indo-Aryan heritage. After removing these overlapping entries, the final filtered Pali lexicon comprised 13,776 unique word forms. This lexicon was then used alongside the Sinhala lexicon in the dual-lexicon classifier described in the following section to assign language labels at the paragraph and page level.

B.4. Corpus Metadata Information

```
{ "BaseURL": "str",
  "Category": "str",
  "Author": {
    "name": "str",
    "DOB": "str",
    "Date_of_passing": "str (YYYY-MM-DD)"
  },
  "book_name_si": "str",
  "book_name_en": "str",
  "number_of_pages": "int",
  "published_date": "str (YYYY-MM-DD)",
  "download_url": "str",
  "post_url": "str",
  "local_path": "str",
  "file_size_mb": "float",
  "sha256_checksum": "str",
  "download_timestamp": "str (ISO 8601)",
  "copyright_analysis": {
    "can_include_in_corpus": "bool",
    "reason": "str",
    "confidence": "str",
    "analysis_date": "str (ISO 8601)"
  }
{
  "extraction": {
    "status": "str",
    "extraction_timestamp": "str (ISO 8601)",
    "extraction_duration_seconds": "float",
    "page_info": {
      "total_pages": "int",
      "text_based_pages": "int",
      "ocr_pages": "int",
      "failed_pages": "int"
    },
    "methods_used": {
      "pdfplumber": "int",
      "document_ai": "int",
      "failed": "int"
    },
    "page_classification": {
      "cover": "int",  "toc": "int",
      "content": "int", "index": "int",
      "blank": "int",  "other": "int"
    },
    "quality_metrics": {
      "ocr_confidence": {
        "avg_confidence": "float",
        "min_confidence": "float",
        "max_confidence": "float",
        "low_confidence_pages": "list[int]",
        "distribution": {
          "high_0.9_to_1.0": "int",
          "medium_0.7_to_0.9": "int",
          "low_below_0.7": "int"
        }
      }
    },
    "file_paths": {
      "raw_text_dir": "str",
      "filtered_text_dir": "str",
      "book_metadata_dir": "str" }}}
```

Figure 5: JSON structure of a metadata record of the books in the IFBC corpus, defining the fields and their corresponding data types.

B.5. Copyright Filtered Books

The list of copyright-filtered books is listed in Table 5.

No.	Title	Author	Published
	Books Related to Tipitaka (5 books)		
1	Maithree Buddha Wanshaya nohoth anagatha wanshaya	Ven. Wilgammula Sangharaja Thero	2015
2	Milind Prashna	Ven. Heenatikubure Sumangala Nahimi	2015
3	Pretha Wasthu	Dhammapala Thero	2015
4	Vimana Wasthu	Dhammapala Thero	2015
5	Wishuddhi Margaya	Buddhaghosa Maha Thero	2015
	Old Buddhist Books (8 books)		
6	Jathila Thotilla	S. Mahinda Thero	2015
7	Pancha Maha Wadaya	Ven. Migettuwatte Gunananda Thero	2015
8	Pali Bhasha Shabdakoshaya	Ven. Widurupola Piyatissa Mahanayaka Thero	2015
9	Sinhala Deepavansaya	Unknown	2015
10	Buthsarana	Vidyachakrawarthi	2015
11	The Book of Dalada Pujavaliya	Unknown	2015
12	The Mahavamsa	Ven. Hikkaduwe Sri Sumangala Ther	2015
13	Saddharma Ratnavali	D.B. Jayatilaka	2015
	Buddhist Characters (3 books)		
14	Ven. Kadawadduwe Jinalankara Thero's Character	IFBC	2015
15	Ven. Man Buridaththa Thero's Character	IFBC	2015
16	Ven. Rerukane Chandawimala Maha Nayaka Thero's Character	IFBC	2015

Table 5: Copyright-cleared historical Buddhist texts included in the corpus (n=16). All books are in the public domain, with authors who passed away before 1954 (70+ years post-mortem).

C. Additional Result Information

C.1. Proprietary Model Evaluation

Proprietary models present a fundamental challenge for perplexity evaluation: production APIs typically do not expose token-level log probabilities required for exact perplexity calculation. Fortunately, OpenAI's API provides access to token-level logprobs through the `logprobs` parameter, enabling precise perplexity measurement for all four evaluated GPT variants.

C.1.1. OpenAI API Logprobs Implementation

For `GPT-3.5-Turbo`, `GPT-4o-mini`, `GPT-4-Turbo`, and `GPT-4o`, we utilise OpenAI's `logprobs` API parameter to extract token-level log probabilities for each input sentence. The evaluation process proceeds as follows:

1. For each test sentence, we make an API call with `logprobs=True` and `top_logprobs=1` to request token-level probability information.

2. The API returns log probabilities for each token in the input sequence (via the `prompt_logprobs` field in API responses where available, or through the `logprobs.content` structure in newer API versions).

3. We extract the log probability $\log P(w_i|w_{<i})$ for each token w_i in the sequence.

4. Perplexity is computed using the standard formula: $\text{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i|w_{<i})\right)$.

5. We implement rate limiting (1-2 second delay between requests) to comply with API usage guidelines and avoid rate limit errors.

This methodology provides exact perplexity measurements for OpenAI models, identical in principle to the evaluation of open-source models with direct access to model internals. The primary difference is computational: API-based evaluation incurs per-token costs and requires network latency for each request, whilst local evaluation provides faster throughput at the cost of computational infrastructure requirements.

C.1.2. Model-Specific Considerations

`GPT-3.5-Turbo`: This provides the baseline for OpenAI's Sinhala capabilities. The API consistently provides logprobs access, enabling reliable perplexity measurement.
`GPT-4o-mini`: This lightweight variant of `GPT-4o` is optimised for cost-efficiency whilst maintaining strong multilingual capabilities. Logprobs access is consistent across API versions.
`GPT-4-Turbo`: The speed-optimised variant of `GPT-4` provides full logprobs access. Note that "Turbo" designation indicates architectural optimisations for inference speed rather than model capacity changes.
`GPT-4o`: The "omni" variant represents OpenAI's latest multimodal architecture with enhanced multilingual capabilities. Full logprobs access is provided, though API response structure differs slightly from earlier `GPT-4` variants (using `logprobs.content` rather than `prompt_logprobs`).

C.2. Proprietary vs Open-Source Models

The perplexity values obtained through OpenAI's API are directly comparable to those from open-source models evaluated locally, as both methods compute the identical mathematical quantity: the exponentiated average negative log-likelihood. The key advantages of API-based evaluation include:

- No local computational infrastructure required
- Access to proprietary models unavailable for local deployment
- Consistent evaluation environment (model version, ttokenisation inference configuration)

The primary disadvantages include:

- Per-token API costs accumulating across 1,024 sentences × 3 corpora
- Network latency and rate limiting, extending evaluation time
- Dependency on API availability and version stability
- Limited transparency regarding model architecture and training data

BNLI: A Linguistically-Refined Bengali Dataset for Natural Language Inference

Farah Binta Haque[1], **Md Yasin**[1,*], **Shishir Saha**[1,*],
Md Shoaib Akhter Rafi[2], **Farig Sadeque**[1]
[1]Brac University, [2]Bangladesh University of Engineering and Technology
[1]Kha 224 Pragati Sarani, Merul Badda, Dhaka 1212, Bangladesh,
[2]ECE Building, West Palashi Campus, Dhaka 1205, Bangladesh
farahhaq2023@gmail.com, yasinislam046@gmail.com, shishirsaha830@gmail.com,
1806114@eee.buet.ac.bd, farig.sadeque@bracu.ac.bd
*These authors contributed equally to this work.

Abstract

Despite the growing progress in Natural Language Inference (NLI) research, resources for the Bengali language remain extremely limited. Existing Bengali NLI datasets exhibit several inconsistencies, including annotation errors, ambiguous sentence pairs, and inadequate linguistic diversity, which hinder effective model training and evaluation. To address these limitations, we introduce BNLI, a refined and linguistically curated Bengali NLI dataset designed to support robust language understanding and inference modeling. The dataset was constructed through a rigorous annotation pipeline emphasizing semantic clarity and balance across entailment, contradiction, and neutrality classes. We benchmarked BNLI using a suite of state-of-the-art transformer-based architectures, including multilingual and Bengali-specific models, to assess their ability to capture complex semantic relations in Bengali text. The experimental findings highlight the improved reliability and interpretability achieved with BNLI, establishing it as a strong foundation for advancing research in Bengali and other low-resource language inference tasks. The link to the BNLI dataset: https://github.com/FarahHaque/BNLI-Dataset.git

Keywords: Natural Language Processing, Natural Language Inference, Bengali Language, Deep Learning

1. Introduction

Natural Language Inference (NLI), also referred to as Recognizing Textual Entailment (RTE), represents a core task in natural language understanding that requires determining whether a given hypothesis can be inferred from a corresponding premise. Over the past decade, large-scale English NLI datasets such as SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2018), and ANLI (Nie and Bansal, 2017) have become foundational benchmarks for advancing deep language models including BERT, RoBERTa, and LLaMA. These resources have substantially contributed to progress in semantic reasoning and contextual understanding across diverse NLP applications (Bowman et al., 2015; Williams et al., 2018; Gururangan et al., 2018). However, comparable resources for low-resource languages—particularly Bengali—remain strikingly underdeveloped.

Although Bengali ranks among the top ten most spoken languages globally, Bengali Natural Language Processing (BNLP) still suffers from a scarcity of high-quality annotated corpora for complex tasks such as entailment recognition (Aggarwal et al., 2022; Bhattacharjee et al., 2022; Kabir et al., 2024). Only a handful of preliminary efforts have been made to construct Bengali NLI datasets; however, most existing corpora suffer from severe shortcomings, including inaccurate translations from English datasets (Aggarwal et al., 2022; Bhattacharjee et al., 2022), inconsistent labeling, syntactic errors, and a lack of semantic and contextual diversity (see Fig. 2). Consequently, models trained on these resources often exhibit limited generalization, biased performance, and poor interpretability when exposed to real-world Bengali text (Kabir et al., 2024; Faria et al., 2024; Mahfuz et al., 2025). This persistent data deficiency has impeded meaningful advancement in Bengali inference modeling and constrained cross-lingual transfer research.

To bridge this critical gap, we present BNLI—a refined and linguistically curated Bengali Natural Language Inference dataset—specifically designed to facilitate robust evaluation and training of NLI models in Bengali. BNLI was developed through a rigorous multi-stage pipeline that combines manual data curation, human validation, and linguistic refinement to ensure both semantic consistency and syntactic authenticity. Unlike earlier datasets, BNLI emphasizes balanced representation across entailment, contradiction, and neutrality classes and incorporates examples reflecting diverse linguistic structures and contextual nuances present in natural Bengali text.

To establish baselines, we evaluated BNLI using several transformer-based architectures, including

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 85–91
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

Table 1: List of existing NLI Datasets on diverse languages including Bangla.

Dataset	Language	Year	Sample Size
SNLI (Bowman et al., 2015)	English	2015	∼570k
MultiNLI (Williams et al., 2018)	English	2017	∼433k
e-SNLI (Camburu et al., 2018)	English	2018	∼570k
ANLI (Nie et al., 2020)	English	2018	∼162k
SICK (Marelli et al., 2014)	English	2019-2020	∼10k
SciTail (Khot et al., 2018)	English	2014	∼27k
RTE (Castillo, 2010)	English	2004-2013	Not Found
SCINLI (Sadat and Caragea, 2022)	English	2022	∼107k
FEVER (Thorne et al., 2018)	English	2018	∼185K
QNLI (Swayamdipta et al., 2020)	English	2018	∼100K
ARABIC-XNLI (Abdelali et al., 2024)	Arabic	2018	∼392k
ARBERT / MARBERT benchmarks (Abdul-Mageed et al., 2021)	Arabic	2021	∼20k
Arabic Natural Language Inference Corpus (Obeidat et al., 2025)	Arabic	2019	∼9k
SANA (Sentiment and NLI Arabic) (Abdul-Mageed and Diab, 2014)	Arabic	2020	∼7k
OSIAN (Open Source International Arabic News) (Zeroual et al., 2019)	Arabic	2019	∼1.1M
ViNLI (Van Huynh et al., 2022)	Vietnamese	2021	∼7.5k
ViHealthNLI (Nguyen et al., 2024)	Vietnamese	2024	∼5k
VietNLI (Bui et al., 2025)	Vietnamese	2023	∼25k
XNLI (Cross-lingual NLI) (Conneau et al., 2018)	Multilingual	2018	∼112k
INDICXNLI (Aggarwal et al., 2022)	Multilingual	2022	∼392k
Multilingual-NLI-26lang-2mil7 (He et al., 2020)	Multilingual	2022	∼2.73M
Cross-Lingual NLI for Low-Resource Languages (MT-NLI) (Zhao, 2022)	Multilingual	2023	∼3.5M
SICK-NL (Wijnholds and Moortgat, 2021)	Dutch	2021	∼10k
NLI-TR (Budur et al., 2020)	Turkish	2020	∼393k
farsTail (Amirkhani et al., 2023)	Persian	2020	∼10k
OCNLI (Original Chinese NLI) (Hu et al., 2020)	Chinese	2020	∼56k
Polish NLI (Ziembicki et al., 2024)	Polish	2022	∼10k
Bangla NLI (BanglaBERT benchmarks) (Bhattacharjee et al., 2022)	Bangla	2022	∼381k

multilingual and Bengali-specific models. The experimental results demonstrate that transformer models outperform recurrent approaches by effectively capturing contextual semantics and long-range dependencies in Bengali. These findings underscore the reliability of BNLI as a benchmark and highlight its potential to accelerate progress in Bengali and other low-resource language inference research (Rashad Al Hasan Rony et al., 2024).

In summary, the main contributions of this study are:

- We develop BNLI, a high-quality and linguistically refined Bengali NLI dataset addressing critical flaws in previous resources.
- We provide comprehensive benchmark evaluations using multiple transformer-based architectures to establish reliable baselines for future research.

By making BNLI publicly available, we aim to promote further exploration in Bengali semantic understanding, encourage resource development for other low-resource languages, and foster inclusive multilingual NLI research.

2. Literature Review

Table 1 presents an overview of major Natural Language Inference (NLI) datasets developed across various languages over the past two decades. As illustrated, extensive and well-curated resources exist for high-resource languages such as English, Arabic, and Chinese, as well as for several emerging multilingual benchmarks like XNLI, INDICXNLI, and MT-based multilingual corpora. These datasets—ranging from early collections such as RTE and SNLI to large-scale multilingual resources—have substantially driven progress in semantic understanding and cross-lingual inference modeling.

In contrast, Bengali remains severely underrepresented in this landscape. Only a limited number of Bengali NLI datasets have been introduced, and most of them are either small in scale or derived from machine-translated versions of English benchmarks. Moreover, many samples in these datasets suffer from semantic inconsistencies, mistranslations, and contextually incorrect entailment labeling, thereby limiting their reliability for model training and evaluation. This evident disparity underscores the urgent need for a linguistically consistent and semantically validated Bengali NLI resource. To address this gap, we introduce BNLI, a refined and meticulously curated dataset aimed at ensuring semantic integrity, syntactic correctness, and balanced class representation for robust Bengali inference modeling.

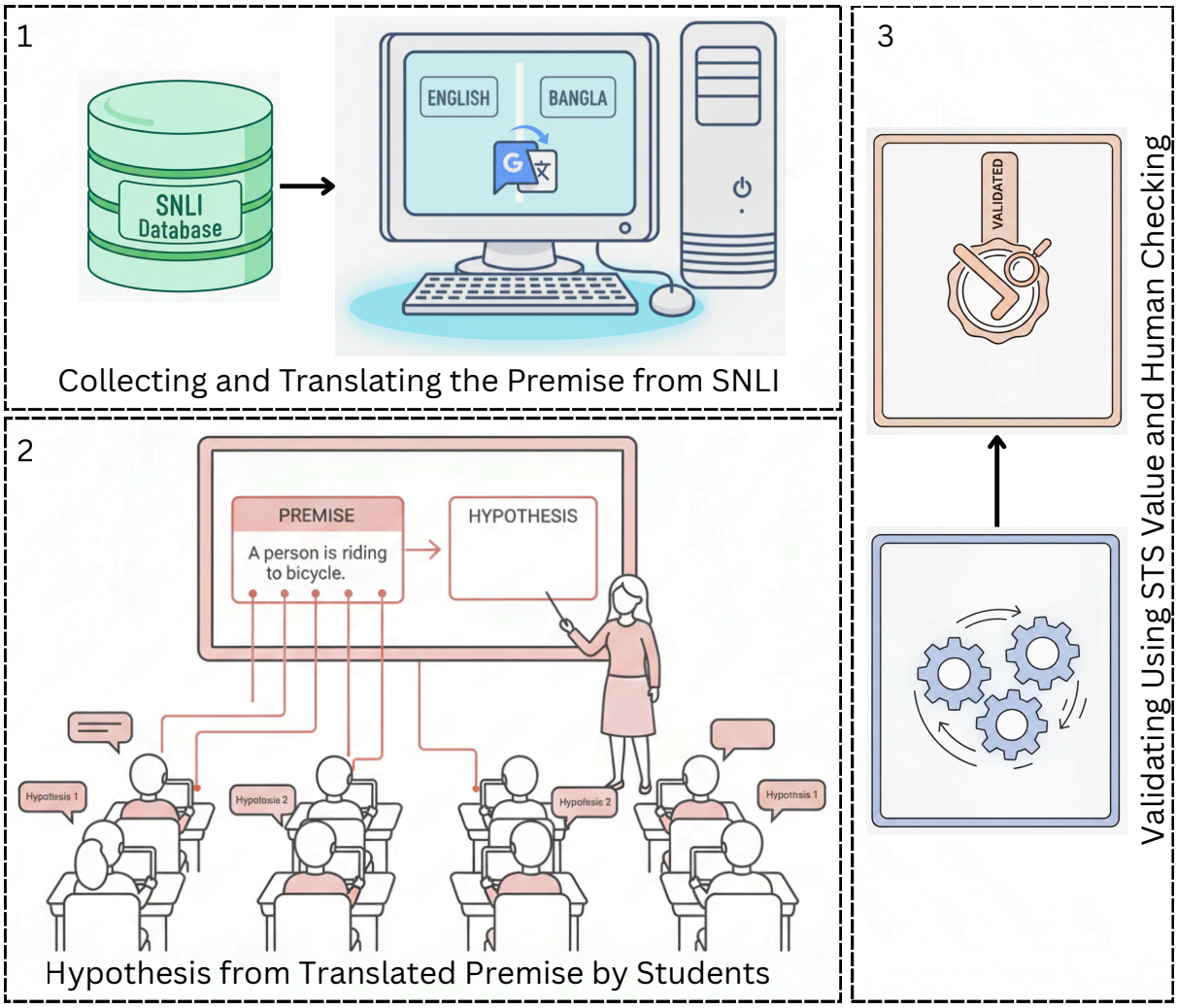


Figure 1: Overall workflow of the proposed multi-stage pipeline adopted for constructing the BNLI dataset, illustrating data collection, linguistic refinement, human validation, and final dataset compilation processes.

Premise	Hypothesis	Label
আমি একটা কেবিন দরজা ভেঙে মাটিতে পড়ে গেলাম-	আমি দরজা ভেঙে পড়ি এবং পড়ে যাই।	entailment
এটা ধীর গতির এটা উহ এই মুহূর্তে বাজারে অনেক ভাল মেশিন আছে	এটাই সবচেয়ে দ্রুতগতির মেশিন, তুমি এর চেয়ে ভালো মেশিন খুঁজে পাবে না।	contradiction
কারন আমি বলতে চাচ্ছি যে শীতের সময় সেখানে কত গরম লাগে সেটা কত	আমি যেখানে থাকি, শীতকালসহ সব সময় গরম থাকে।	neutral

Figure 2: Sample entries from the Bangla NLI dataset, showing grammartically inconsistent premise-hypothesis pairs with entailment, neutral, and contradiction labels.

3. Methodology

The BNLI dataset was constructed through a three-stage data collection and refinement process to ensure linguistic accuracy and semantic reliability. The data collection process is illustrated in Fig. 1.

In the first stage, 8885 sentences were selected as premises from the SNLI corpus (Bowman et al., 2015). These sentences were translated into Bengali with careful attention to preserving grammatical integrity and natural sentence flow. In the second stage, we focused on constructing semantically aligned hypotheses for the collected Bengali premises. To generate corresponding hypotheses, 60 native Bengali-speaking students from a reputed university participated in a structured data collection process. Each participant was provided with premise–scene descriptions and tasked with composing hypothesis sentences through Google

Premise	Hypothesis	Label	Similarity
শিশুরা হাসতে হাসতে ক্যামেরার দিকে হাত নাড়াচ্ছিল ।	শিশুরা হাসিমুখে ক্যামেরার দিকে তাকিয়ে হাত নাড়াচ্ছিল ।	entailment	0.923020
শিশুরা হাসতে হাসতে ক্যামেরার দিকে হাত নাড়াচ্ছিল ।	মেয়েরা গম্ভীর মুখে তার কথা ভিডিও করছিল ।	contradiction	0.110095
শিশুরা হাসতে হাসতে ক্যামেরার দিকে হাত নাড়াচ্ছিল ।	শিশুরা হাসতে হাসতে খেলা করছিল ।	neutral	0.622482
একজন বৃদ্ধ লোক একটি রেস্তরায় বসে কমলার জুস পান করছিল ।	একজন বৃদ্ধ লোক জুস খাচ্ছে ।	entailment	0.896547
একজন বৃদ্ধ লোক একটি রেস্তরায় বসে কমলার জুস পান করছিল ।	দুই মহিলা একটি রেস্তোরাঁয় মদ খাচ্ছেন ।	contradiction	0.182121
একজন বৃদ্ধ লোক একটি রেস্তরায় বসে কমলার জুস পান করছিল ।	একজন লোক একটি দোকানে জুস খাচ্ছেন ।	neutral	0.721758
দুই স্বর্ণকেশী মহিলা একে অপরকে আলিঙ্গন করছে ।	দুইজন মহিলা একে অপরকে আলিঙ্গন করছে ।	entailment	0.927653
দুই স্বর্ণকেশী মহিলা একে অপরকে আলিঙ্গন করছে ।	দুই কিশোর একসাথে খেলছে ।	contradiction	0.153298
দুই স্বর্ণকেশী মহিলা একে অপরকে আলিঙ্গন করছে ।	কিশোরী দুইজন একে অপরকে আপ্যায়ন করছে ।	neutral	0.684532

Figure 3: Sample entries from the BNLI dataset, showing premise-hypothesis pairs with entailment, neutral, and contradiction labels, along with estimated STS scores.

Algorithm 1 STS-Based Filtering

Require: Premise–hypothesis pair (p, h) and label $y \in \{\text{Contr.}, \text{Neut.}, \text{Ent.}\}$
Ensure: Decision $d \in \{\text{Keep, Drop}\}$
Translate (p, h) to English
Compute STS score $s \in [0, 1]$ using *en_stsb_bert_large* of Spacy
Set $d \leftarrow$ Drop
if $(y = \text{entailment}$ **and** $s > 0.70)$ **or** $(y = \text{neutral}$ **and** $0.30 < s < 0.75)$ **or** $(y = \text{contradiction}$ **and** $s < 0.30)$ **then**
    $d \leftarrow$ Keep
end if
return d

Forms. This process resulted in 26655 sentence pairs, evenly distributed across the three NLI labels: entailment, contradiction, and neutral.

In the third stage, all collected sentence pairs underwent a comprehensive validation process combining Semantic Textual Similarity (STS) analysis and human evaluation. Each premise–hypothesis pair was translated into English using the Google Translator Python package, and STS scores were computed to assess semantic alignment with the assigned NLI labels using *en_stsb_bert_large* model of Spacy library to determine STS value within 0 to 1. The premise-hypothesis dropping criteria is as described in Algorithm 1.

Pairs with inconsistent or outlier scores were flagged for manual inspection. Subsequently, five native Bengali linguists reviewed these pairs to verify grammatical correctness, semantic coherence, and label validity, resolving discrepancies through consensus. The final dataset contains total 23067 refined sentence pairs (Entailment: 7682 pairs, Contradiction: 7696 pairs, Neutral: 7661 pairs). This dual validation approach ensured that the BNLI dataset maintained high linguistic fidelity, semantic consistency, and balanced representation across inference categories. Arbitrary samples of the premise and hypothesis pairs from BNLI dataset has been shown in Fig. 3.

4. Result Analysis

The results on the BNLI dataset (Table. 2) show that transformer-based models substantially outperform traditional architectures. The LSTM baseline performs poorly, reflecting its limited ability to capture semantic nuances. BERT Base and RoBERTa deliver strong and balanced results, with RoBERTa slightly ahead due to improved contextual representations. BanglaBERT shows inconsistent performance across classes, suggesting limitations in domain coverage. MultiBERT performs reasonably well, benefiting from multilingual training. The LLaMA-2 model achieves the highest overall F1-score (79.13%), demonstrating superior cross-lingual reasoning and robust generalization compared to all other models.

Table 2: Model-wise performance comparison on BNLI dataset.

Model	#Params(M)	Classwise Performance			Average Performance			
		Contr.	Neut.	Ent.	Acc.	Prec.	Rec.	F1
LSTM	∼10	45.12%	46.38%	41.57%	44.19%	45.06%	44.32%	44.68%
BERT Base	110	76.42%	73.19%	64.87%	70.90%	72.45%	71.66%	71.92%
BERT Large	340	83.18%	68.73%	53.24%	67.5%	68.12%	68.44%	68.27%
BanglaBERT	168	32.46%	11.58%	88.12%	45.4%	49.37%	48.73%	44.26%
RoBERTa	125	79.68%	74.91%	67.43%	69.66%	74.65%	73.92%	74.21%
MultiBERT	185	86.23%	75.62%	48.91%	68.75%	69.58%	70.34%	69.93%
LLaMA-2	7,000	**84.76%**	**80.35%**	**72.41%**	**74.56%**	**79.51%**	**78.92%**	**79.13%**

The strong performance of LLaMA-2, although unexpected for NLI tasks on relatively small datasets, can be attributed to several factors. Its large-scale pretraining on diverse corpora enables implicit learning of semantic and inference patterns, while instruction-following and contextual reasoning capabilities allow it to interpret premise–hypothesis relationships effectively without extensive task-specific fine-tuning. Additionally, our STS-based filtering strategy produces semantically well-aligned and less noisy pairs, which may favor models with strong semantic similarity understanding, helping LLaMA-2 outperform smaller, fine-tuned encoders such as BERT variants. In contrast, BanglaBERT exhibits substantially lower performance, particularly for the neutral class, due to the inherent ambiguity of neutral samples, potential class imbalance, and dataset-specific characteristics that differ from prior benchmarks such as Bhattacharjee et al. This discrepancy further suggests that domain differences and implementation-specific factors (e.g., fine-tuning strategy, preprocessing, or tokenization) can significantly influence results, highlighting that previously reported performance may not directly generalize to the BNLI dataset.

5. Conclusion

In this work, we presented BNLI, a high-quality Bengali NLI dataset addressing the limitations of existing resources, including annotation inconsistencies and linguistic imbalance. Through a careful curation and annotation process, BNLI ensures clear semantic distinctions across entailment, contradiction, and neutral classes, making it suitable for reliable model evaluation and training. Our extensive benchmarking with both multilingual and Bengali-specific transformer models demonstrates the dataset's effectiveness in revealing model strengths and weaknesses, as well as the substantial performance gains achievable with large-scale language models such as LLaMA-2. Overall, Bengali NLI dataset serves as a foundational resource for advancing natural language understanding in a low-resource setting. A key future direction is the development of a Bengali foundation model, leveraging this dataset for large-scale pretraining and downstream reasoning tasks. The dataset supports applications in natural language inference and can further be utilized to train LLMs for meaningful sentence generation in the Bengali language.

6. References

Ahmed Abdelali, Hamdy Mubarak, Shammur Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Samir Abdaljalil, Yassine El Kheir, Daniel Izham, Fahim Dalvi, et al. 2024. Larabench: Benchmarking arabic ai with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–520.

Muhammad Abdul-Mageed and Mona T Diab. 2014. Sana: A large scale multi-genre, multi-dialect lexicon for arabic subjectivity and sentiment analysis. In *LREC*, pages 1162–1169.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, et al. 2021. Arbert & marbert: Deep bidirectional transformers for arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105.

Divyanshu Aggarwal, Vivek Gupta, and Anoop Kunchukuttan. 2022. Indicxnli: Evaluating multilingual inference for indian languages. *arXiv preprint arXiv:2204.08776*.

Hossein Amirkhani, Mohammad AzariJafari, Soroush Faridan-Jahromi, Zeinab Kouhkan, Zohreh Pourjafari, and Azadeh Amirak. 2023.

Farstail: A persian natural language inference dataset. *Soft Computing*, pages 1–13.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 632–642.

Emrah Budur, Rıza Özçelik, Tunga Güngör, and Christopher Potts. 2020. Data and representation for turkish natural language inference. *arXiv preprint arXiv:2004.14963*.

Tran Bao Bui, Linh Thi-Thuy Nguyen, and Tin Van Huynh. 2025. Vietx-nli: A cross-lingual natural language inference dataset with vietnamese as the source language. In *2025 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pages 1–6. IEEE.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Julio J Castillo. 2010. Recognizing textual entailment: experiments with machine learning algorithms and rte corpora. *Special issue: Natural Language Processings and its Applications, Research in Computing Science*, 46:155–164.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2475–2485.

Fatema Tuj Johora Faria, Mukaffi Bin Moin, Asif Iftekher Fahim, Pronay Debnath, and Faisal Muhammad Shah. 2024. Unraveling the dominance of large language models over transformer models for bangla natural language inference: A comprehensive study. In *International Conference on Computing and Communication Networks*, pages 13–24. Springer.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence S Moss. 2020. Ocnli: Original chinese natural language inference. *arXiv preprint arXiv:2010.05444*.

Mohsinul Kabir, Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Mir Tafseer Nayeem, M Saiful Bari, and Enamul Hoque. 2024. Benllm-eval: A comprehensive evaluation into the potentials and pitfalls of large language models on bengali nlp. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2238–2252.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Tamzeed Mahfuz, Satak Kumar Dey, Ruwad Naswan, Hasnaen Adil, Khondker Salman Sayeed, and Haz Sameen Shahgir. 2025. Too late to train, too early to use? a study on necessity and viability of low-resource bengali llms. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1183–1200.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 1–8.

Huyen Nguyen, Thanh-Ha Do, Tuan-Anh Hoang, et al. 2024. Vihealthnli: A dataset for vietnamese natural language inference in healthcare. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages@ LREC-COLING 2024*, pages 404–409.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. *arXiv preprint arXiv:1708.02312*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4885–4901.

Rasha Obeidat, Yara Al-Harahsheh, Mahmoud Al-Ayyoub, and Maram Gharaibeh. 2025. Arentail: manually-curated arabic natural language inference dataset from news headlines. *Language Resources and Evaluation*, 59(1):509–535.

Md Rashad Al Hasan Rony, Sudipto Kumar Shaha, Rakib Al Hasan, Sumon Kanti Dey, Amzad Hossain Rafi, Amzad Hossain Rafi, Ashraf Hasan Sirajee, and Jens Lehmann. 2024. Banglaquad: A bengali open-domain question answering dataset. *arXiv e-prints*, pages arXiv–2410.

Mobashir Sadat and Cornelia Caragea. 2022. Scinli: A corpus for natural language inference on scientific text. *arXiv preprint arXiv:2203.06728*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *arXiv preprint arXiv:2009.10795*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Tin Van Huynh, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. 2022. Vinli: A vietnamese corpus for studies on open-domain natural language inference. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3858–3872.

Gijs Wijnholds and Michael Moortgat. 2021. Sicknl: A dataset for dutch natural language inference. *arXiv preprint arXiv:2101.05716*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, pages 1112–1122.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. Osian: Open source international arabic news corpus-preparation and integration into the clarin-infrastructure. In *Proceedings of the fourth arabic natural language processing workshop*, pages 175–182.

Jiawei Zhao. 2022. *Cross-lingual learning in low-resource: a thesis presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science, School of Natural and Computational Sciences, Massey University, Auckland, New Zealand*. Ph.D. thesis, Massey University.

Daniel Ziembicki, Karolina Seweryn, and Anna Wróblewska. 2024. Polish natural language inference and factivity: An expert-based dataset and benchmarks. *Natural Language Engineering*, 30(2):385–416.

Exploring Large Language Models for Multitask Learning in Bengali Text Classification

Md. Sajjad Hossain[1], **Kawsar Ahmed**[1], **Suny Md Ashraf Khan**[2]
Mohammed Moshiul Hoque[1]
[1]NLP Lab, Chittagong University of Engineering and Technology, Bangladesh,
[2]Infratech Consultants Ltd, Bangladesh
u1904031@student.cuet.ac.bd, ashraf@infraconsultants.com, moshiul_240@cuet.ac.bd

Abstract

Text classification in low-resource languages has become increasingly important due to the rapid growth of user-generated digital content. While multitask learning has long been studied in NLP, the use of LLMs for multitask text classification in low-resource languages such as Bengali remains underexplored. Although LLMs are inherently multilingual and multitasking, their effectiveness in structured multitask classification settings for Bengali has not been systematically evaluated. In this work, we investigate how LLMs can be leveraged for multitask Bengali text classification across five domains: sentiment analysis, aggressive text detection, fake news detection, news categorization, and emotion analysis. We compare in-context learning strategies—including zero-shot, one-shot, and chain-of-thought prompting—with parameter-efficient fine-tuning approaches. Our findings show that CoT prompting does not consistently improve performance and often degrades performance, highlighting the instability of prompt-based adaptation in low-resource settings with limited pretraining exposure. Moreover, reasoning-optimized models such as DeepSeek-R1 exhibit substantial performance drops, indicating that enhanced reasoning capabilities alone cannot overcome the challenges posed by low-resource settings. Among the evaluated mLLMs, Gemma-3-4B demonstrates the most stable and balanced cross-task performance under both in-context learning and parameter-efficient fine-tuning, making it a strong backbone candidate for multitask Bengali text classification. These results provide empirical evidence on the limitations of prompting and the advantages of lightweight fine-tuning for low-resource multilingual NLP.

Keywords: Natural Language Processing, LLMs, Low-Resource Languages, Multitask Text Classification

1. Introduction

With the rapid expansion of internet access and social media usage in Bangladesh and Bengali-speaking communities, large volumes of user-generated textual content are produced daily. This content often contains sentiment, misinformation, aggression, and emotionally expressive language, making automated text classification (TC) an essential tool for content moderation, information verification, and social media analysis. In recent years, several studies have addressed Bengali TC tasks individually, including sentiment analysis (Bhowmick and Jana, 2021), fake news detection (Farhad et al., 2024), emotion classification (Das et al., 2023), and aggressive text detection (Rosni et al., 2024). These works have contributed valuable task-specific datasets and modeling approaches. However, most existing research focuses on single-task settings, where models are trained independently for each classification objective. In contrast, multitask learning (MTL)—which enables a unified model to learn shared representations across multiple related tasks—remains relatively underexplored in Bengali NLP. Although multi-task text classification has witnessed significant advancements in high-resource languages such as English and Chinese, progress in low-resource languages—including Bengali—remains limited (Afroz et al.). The development of robust Bengali classification systems is hindered by the scarcity of standardized annotated corpora, limited domain-diverse datasets, insufficient pre-trained embeddings, and comparatively underdeveloped NLP infrastructure. These challenges are further compounded in multitask settings, where a single model must generalize across heterogeneous domains with varying linguistic and semantic complexities.

Recent advances in Large Language Models (LLMs) have transformed NLP by enabling deep contextual understanding, cross-lingual transfer, and strong zero-shot and few-shot performance. LLMs have demonstrated impressive results across various classification tasks and languages (Ding et al., 2023; Wang et al., 2025). Their multilingual pretraining suggests potential applicability to low-resource languages. Despite this promise, the effectiveness of LLMs for structured multitask text classification in Bengali remains largely unexplored. In particular, there is limited empirical evidence comparing different adaptation strategies—such as in-context learning and parameter-efficient fine-tuning—for Bengali classification tasks. Although modern LLMs are multilingual, their pretraining corpora are disproportionately dominated by high-resource lan-

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 92–104
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

guages. This imbalance often leads to suboptimal performance in low-resource contexts, where linguistic structures, morphology, cultural expressions, and script characteristics differ significantly from those of dominant languages. Understanding how different LLM architectures and adaptation techniques perform under these constraints is therefore essential for advancing multilingual NLP research.

In this work, we systematically investigate the use of LLMs for multitask Bengali text classification across multiple domains. By evaluating different adaptation paradigms and analyzing their task-specific behavior, we aim to provide empirical insights into the capabilities and limitations of LLMs in low-resource multilingual settings. The key findings of this work can be summarized as follows:

- The results demonstrate that Chain-of-Thought (CoT) prompting does not consistently enhance performance in Bengali text classification and, in several cases, leads to measurable degradation. This finding underscores the instability of prompt-based adaptation strategies in low-resource settings where pretraining exposure to the target language is limited.
- Reasoning-optimized large language models, such as DeepSeek-R1, exhibit significant performance deterioration across multiple tasks. This reveals a critical limitation of scaling reasoning capacity without sufficient linguistic grounding, particularly in low-resource language contexts.
- Among the evaluated models, Gemma-3-4B achieves the most stable and consistent performance across diverse tasks under both in-context learning and parameter-efficient fine-tuning. Its balanced cross-task generalization highlights its suitability as a backbone model for multitask Bengali text classification.

2. Related Work

There has been significant progress on TC tasks in high-resource languages (HRLs) using LLMs. In contrast, low-resource languages (LRLs) like Bengali have seen much less improvement.

2.1. Text Classification in HRLs

There are numerous ways of text classification using ML, DL, transformers, and LLMs. Most research on LLMs has focused on high-resource languages, such as English. Alex et al., 2021 proposed a few-shot text classification benchmark using GPT-3. They evaluated 11 tasks across 11 datasets in English. Schick and Schütze, 2022 also performed 11 tasks, as before, and used the same dataset for each task. They used Pattern-Exploiting Training (PET), which performed nearly as well as non-expert humans in 7 out of 11 tasks. Their model outperformed that of Alex et al., 2021 on several tasks. However, they could not improve the accuracy of some tasks, which appeared to be very low. Loukas et al., 2023 used context learning with GPT-3.5 and GPT-4 for classifying text of Banking77 datasets. This dataset comprises 77 classes and 13083 samples. Their approach outperformed fine-tuned, non-generative models even with fewer samples. They got an F1 score of 0.83. Wang et al., 2024 proposed a simplified approach for classifying text with LLMs (GPT-4, GPT-3.5, Gemini-pro, Llama-3-8b, etc) using zero and few-shot prompting. They demonstrated that zero-shot prompting is a more convenient approach for text classification than traditional methods (ML, DL, and Transformer models). Although several TC methods in HRLs have demonstrated promising results, these models cannot be used for resource-constrained languages (e.g., Bengali) due to insufficient domain-specific training data, linguistic diversity, fine-tuned embeddings, and classification methods.

2.2. Text Classification in LRLs

In addition to HRLs, there has been extensive research on LRLs such as Telugu, Marathi, and Tamil. Marreddy et al., 2022 proposed MT-Text GCN, which combines graph autoencoder-based graph reconstruction and multi-task text classification in Telugu. They achieved F1 scores of 0.84 (Sentiment), 0.55 (Emotion), 0.83 (Hate Speech), and 0.66 (Sarcasm). Zhang et al., 2021 also proposed a multi-task learning framework for sentiment analysis on various datasets, combining hard parameter sharing, a BERT-based shared encoder, and a task recognition mechanism. Their approach achieved an F1 score of 0.91. Kapil and Ekbal, 2025 proposed a novel multi-task learning framework for detecting hateful memes. They integrated three pre-trained models: CLIP, BERT, and UNITER, as their work involves both text and images. On the MultiOFF and MAMI datasets, they achieved F1 scores of 0.79 and 0.84, respectively. A recent study (Singh et al., 2024) proposed a multimodal zero-shot multi-task learning framework that uses cross-attention to identify intent and emotion. Their model achieved an increase in F1 score of 0.06 for intent detection and 0.04 for emotion detection compared to other approaches.

2.2.1. Text Classification in Bengali

In contrast to the other languages, TC research leveraging LLMs in Bengali is currently at a rudimentary stage. A limited number of studies also utilized LLMs for TC tasks. For example, Kabir et al., 2023 conducted multiple TC tasks such as question answering, paraphrasing, summarization, and text classification using LLMs (GPT-3.5, Claude-2, LLaMA-2). All the LLMs performed poorly with an F1 score of 0.49 (Claude-2), 0.48 (GPT-3.5), and 0.29 (LLaMA-2). Hasan et al., 2023 explored LLMs with zero and few-shot techniques for Bengali sentiment analysis. The performance of GPT-4 was inferior in their study. The F1 score of GPT-4 in various settings was 0.60 (0-shot), 0.60 (3-shot), and 0.61 (5-shot). A recent work (Nazi et al., 2025) explored various LLMs across four TC tasks, including text classification, sentiment analysis, summarization, and question answering. Their results demonstrated that the closed-source GPT-4o model, utilizing few-shot learning and chain-of-thought prompting, achieved the highest performance across multiple tasks. Few studies also focused on TC in various domains using ML and DL techniques, such as sports news classification (Barua et al., 2021), suspicious text classification (Sharif et al., 2020), and authorship classification (Hossain et al., 2021b). In recent days, transformer-based approaches have gained much attention in various Bengali TC research, including emotion classification (Das et al., 2023; Hider et al., 2024), sentiment analysis (Ahsan et al., 2023), technical text classification (Aodhora and Hoque, 2024), fake news classification (Akther et al., 2025), authorship attribution (Hossain et al., 2025), and aggressive text classification (Sharif and Hoque, 2022). Past research has shown that Bengali TC models struggle to comprehend new tasks that are not widely recognized.

Most past studies in Bengali have focused on a single task or domain, lacking a unified evaluation benchmark for multiple TC tasks or domains. To address this gap, this study explores five distinct Bengali text classification tasks using LLMs.

3. Datasets

The proposed method is evaluated on five Bengali tasks: Sentiment Analysis (SA), Aggressive Text Detection (ATD), Fake News Detection (FND), News Headline Categorization (NHC), and Textual Emotion Analysis (TEA). Each task utilizes its respective dataset from available sources, including SA (Hossain et al., 2021a), ATD (Sharif et al., 2022), FND (Hossain et al., 2020), NHC (Hossain, 2023), and TEA (Das et al., 2023). For in-context learning, the evaluation dataset is used directly without modification. However, for PEFT (Ding et al., 2023), changes were made by manually adding task-specific instructions (prompts) to help the model identify the task being addressed. Additionally, to reduce output ambiguity, a class label name and a corresponding number were added to each output. Table 1 presents a summary of all datasets used in this study. This study encompasses 6,026 Bengali text samples across five distinct classification tasks for evaluation.

For instruction tuning, we add task-specific samples to create a comprehensive training set. Each sample includes detailed task instructions, specific prediction guidance, and class names with corresponding labels for clarity and accuracy. Table 2 provides a summary of the instruction tuning dataset composition. The instruction tuning dataset is a significant expansion over the evaluation datasets, comprising 32,561 samples. To create an instruction dataset, we first inject a prompt before each sample in each task. The prompt provides an instruction to determine the accurate class of the task.

Table 3 provides an overview of the Alpaca-style instructions dataset for the tasks. The alpaca-style dataset guides the model to produce outputs in the desired format for Bengali classification tasks. In contrast, untuned LLMs often generate extraneous text. By training on this dataset, the model is constrained to provide concise, relevant responses tailored to the target classification task. Figure 1 illustrates the distribution of classes among all tasks in instruction tuning and evaluation datasets. It shows that the distributions are nearly identical across the two sets. Among all tasks, NHC has the most data (15,000) across six classes, whereas SA has the least (1,010) across two classes. In NHC, 35.9% of the samples belong to the International category, while only 1.9% of the samples belong to the IT category in the instruction tuning datasets.

4. System Overview

This study explores both in-context learning and PEFT techniques to perform the multiple downstream tasks. Figure 2 provides an overview of the overall system.

4.1. Models

We have explored eight LLMs to evaluate five Bengali text classification tasks. Each model varies according to their internal settings. We have utilized Llama-3.2 3B (Grattafiori et al., 2024), Mistral-7B (**?**), Qwen-2.5 (Yang et al., 2025), Phi-3 (Abdin et al., 2024), Deepseek-7B (Bi et al., 2024), Gemma-3 (Team et al., 2025) and DeepSeek-R1

Task	Data	LR	T_S	T_W	Classes	Class Labels
SA (Hossain et al., 2021a)	434	1-562	456	10,909	2	Positive, Negative
ATD (Sharif et al., 2022)	1,416	3-595	1897	31189	2	Aggressive, Non-Aggressive
FND (Hossain et al., 2020)	2,551	390-3,761	54,508	6,67,351	2	Fake, Authentic
NHC (Hossain, 2023)	1,000	2-9	1000	5932	6	Politics, Sports, National, Entertainment, International, IT
TEA (Das et al., 2023)	625	4-97	832	14,508	6	Joy, Sadness, Fear, Disgust, Anger, Surprise
Total	6,026	-	58,693	729,889	-	-

Table 1: Overview of Bengali text classification evaluation datasets. Here, LR refers to the range of minimum and maximum lengths of each sample, while T_s and T_w denote the total number of sentences and total number of words.

Task	Samples	T_S	T_W
SA	1,010	999	25,443
ATD	10,000	13,491	2,11,096
FND	2,551	1,26,004	1554914
NHC	15,000	15,000	88,512
TEA	4,000	6543	1,14,674
Total	32,561	162,037	1,994,639

Table 2: Instruction tuning dataset statistics for tuning LLMs.

(Guo et al., 2025). For model implementation, we used 4-bit quantized models and QLoRA to tune instructions under limited GPU resources efficiently. Larger closed-source models like DeepSeek-R1 was accessed via their APIs. All other models, ranging from 3 billion to 7 billion parameters, were accessed and executed locally within the Kaggle environment.

4.2. In-Context Learning (ICL)

For ICL, we used different prompting techniques like zero-shot, one-shot, and CoT across five Bengali text classification tasks.

Appendix A illustrates the prompts used in this study.

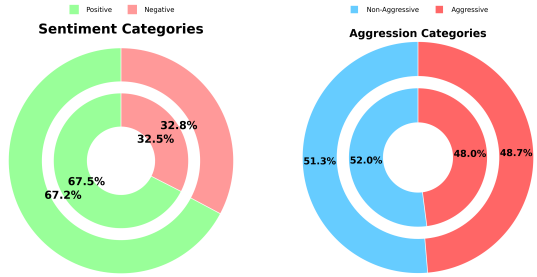
- **Zero-shot Prompt:** The model is given only the task description without any examples.
- **One-shot Prompt:** The model is provided with a single example of the task along with its answer.
- **Chain-of-Thought (CoT) Prompt:** The model is first given the task context, and then guided to reason through the problem step by step, rather than producing a direct answer.

Instruction	Input	Output
Find the sentiment of the following Bengali sentence.	এই সিনেমাটা অসাধারণ ছিল। (This movie was amazing.)	Positive
Detect whether the following Bengali sentence is aggressive or not.	তুই একটা অকাজের গাধা! (You're a useless donkey!)	Aggressive
Classify whether the following Bengali news is fake or real.	প্রধানমন্ত্রী বলেছেন যে আগামীকাল ৫০০০ টাকা করে দেওয়া হবে। (The Prime Minister said that 5,000 taka will be given to each person tomorrow.)	Fake
Categorize the topic of the following Bengali news headline.	তাদের মিটিং-মিছিলের জন্য দেশ স্বাধীন করা হয়নি: হানিফ। (The country was not liberated for their meetings and rallies: Hanif.)	Politics
Detect the emotion expressed in the following Bengali sentence.	আজ আমার খুব কষ্ট হচ্ছে। (I'm feeling very sad today.)	Sad

Table 3: Sample instructions, inputs, and outputs for each task.

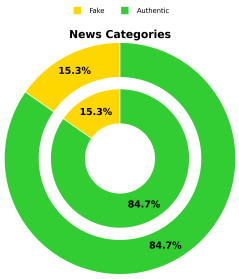
4.3. Parameter Efficient Fine-tuning (PEFT)

In this approach, we have utilized different adapters for each task. We have trained five sepa-

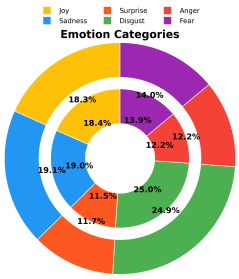


(a) SA

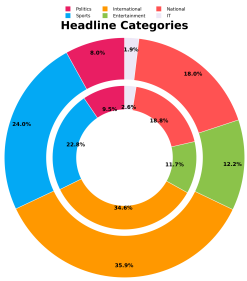
(b) ATD



(c) FND



(d) TEA



(e) NHC

Figure 1: Class-wise distribution of data among all tasks in train and test set.

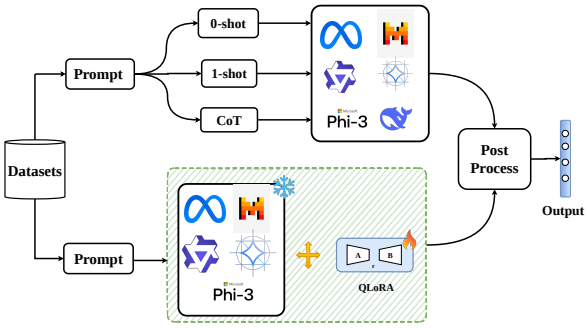


Figure 2: Schematic Diagram of LLM Evaluation for Bengali TC Tasks

rate QLoRA (Dettmers et al., 2023) adapters, one for each of the five tasks, using the SA, ADT, FND, NHC, and TEA datasets. Formally, for a transformer layer weight matrix $W \in \mathbb{R}^{d\times k}$, the adapted weight W' is computed as:

$$W' = W + \Delta W, \quad \Delta W = AB^\top \tag{1}$$

where $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{k\times r}$ are trainable low-rank matrices with rank $r \ll \min(d, k)$. In our implementation, we set $r = 16$, which provides a balance between parameter efficiency and expressive capacity.

Following the LoRA formulation, the low-rank update is scaled as:

$$\Delta W = \frac{\alpha}{r} AB^\top \tag{2}$$

where α is a scaling factor. We set $\alpha = 32$ to ensure a stable update magnitude while preserving the benefits of low-rank adaptation. To improve memory efficiency, we adopt 4-bit quantization for the base model weights, following the QLoRA framework. This allows training large language models under limited hardware constraints by storing backbone weights in low-precision while maintaining adapter parameters in high-precision. We set the dropout rate to 0 based on empirical tuning, as preliminary experiments showed that additional dropout did not improve performance. During inference, the base model remains fixed, and the appropriate task-specific adapter is dynamically loaded.

After generating an output, the LLM may include unnecessary text, which is removed during post-processing using regular expressions. Also, if the responses were found in the wrong format after post-processing, then they were treated as wrong answers.

4.4. Experimental Setup

This study was conducted using Python 3.10.12, PyTorch 2, and Unsloth to implement large language models (LLMs). Experiments were run on Kaggle using an NVIDIA Tesla P100 GPU, which requires 29 GB of RAM, 16 GB of VRAM, and 73.1 GB of storage. Unsloth was used for PEFT with QLoRA, while DeepSeek-70B APIs were employed. We evaluated performance using three prompting strategies and the metrics: precision (Pr), recall (Re), F1-score (F1), and accuracy (Acc).

5. Results and Analysis

Table 4 demonstrates the evaluation results of eight LLMs across five tasks.

How do the models behave in different ICL settings? Across zero-shot, one-shot, and chain-of-thought (CoT) prompting settings, no single model consistently outperforms others across all

Task	Model	Zero-shot				One-shot				CoT			
		Pr	**Re**	**F-1**	**Acc**	**Pr**	**Re**	**F-1**	**Acc**	**Pr**	**Re**	**F-1**	**Acc**
SA	Llama	0.93	0.92	0.93	0.94	0.97	0.95	0.96	0.97	0.97	0.95	0.94	0.94
	Gemma	0.93	0.84	0.88	0.89	0.91	0.89	0.90	0.88	0.87	0.89	0.88	0.86
	Mistral	0.70	0.68	0.69	0.71	0.65	0.67	0.66	0.64	0.64	0.67	0.66	0.68
	Qwen	**0.96**	**0.94**	**0.95**	**0.95**	**0.97**	**0.95**	**0.96**	**0.94**	**0.97**	**0.95**	**0.96**	**0.94**
	Phi	0.71	0.73	0.72	0.70	0.64	0.67	0.66	0.68	0.65	0.67	0.66	0.64
	Dseek7B	0.81	0.79	0.80	0.83	0.84	0.82	0.83	0.85	0.84	0.82	0.83	0.85
	DseekR1	0.88	0.90	0.89	0.87	0.89	0.91	0.90	0.88	0.89	0.91	0.90	0.88
ATD	Llama	**0.81**	**0.83**	**0.82**	**0.84**	0.72	0.61	0.54	0.59	0.71	0.69	0.70	0.72
	Gemma	0.81	0.79	0.80	0.82	**0.78**	**0.76**	**0.77**	**0.79**	**0.77**	**0.75**	**0.76**	**0.78**
	Mistral	0.60	0.63	0.62	0.61	0.62	0.58	0.54	0.57	0.60	0.62	0.61	0.59
	Qwen	0.70	0.68	0.69	0.71	0.66	0.64	0.65	0.67	0.71	0.69	0.70	0.72
	Phi	0.50	0.52	0.51	0.49	0.51	0.49	0.50	0.52	0.50	0.48	0.49	0.51
	Dseek7B	0.58	0.60	0.59	0.57	0.58	0.56	0.57	0.59	0.45	0.43	0.44	0.46
	DseekR1	0.80	0.78	0.77	0.77	0.63	0.61	0.62	0.64	0.43	0.41	0.42	0.44
FND	Llama	0.62	0.70	0.63	0.73	0.61	0.69	0.48	0.51	0.77	0.75	0.76	0.78
	Gemma	**0.87**	**0.89**	**0.88**	**0.86**	**0.85**	**0.83**	**0.84**	**0.86**	**0.79**	**0.77**	**0.78**	**0.80**
	Mistral	0.60	0.54	0.49	0.58	0.50	0.48	0.49	0.51	0.49	0.47	0.48	0.50
	Qwen	0.70	0.68	0.69	0.71	0.64	0.62	0.63	0.65	0.74	0.72	0.73	0.75
	Phi	0.56	0.54	0.55	0.57	0.43	0.41	0.42	0.44	0.64	0.62	0.63	0.65
	Dseek7B	0.45	0.43	0.44	0.46	0.45	0.43	0.44	0.46	0.42	0.40	0.41	0.43
	DseekR1	0.92	0.93	0.87	0.83	0.67	0.65	0.66	0.68	0.69	0.67	0.68	0.70
TEA	Llama	0.61	0.42	0.32	0.33	0.63	0.58	0.52	0.53	0.48	0.47	0.45	0.46
	Gemma	**0.57**	**0.56**	**0.55**	**0.64**	**0.66**	**0.63**	**0.57**	**0.59**	**0.56**	**0.54**	**0.49**	**0.50**
	Mistral	0.51	0.31	0.25	0.29	0.55	0.39	0.29	0.36	0.34	0.30	0.24	0.30
	Qwen	0.51	0.45	0.42	0.44	0.46	0.44	0.37	0.39	0.51	0.51	0.47	0.48
	Phi	0.47	0.40	0.37	0.39	0.30	0.25	0.24	0.25	0.30	0.28	0.24	0.31
	Dseek7B	0.30	0.28	0.23	0.29	0.32	0.30	0.27	0.29	0.35	0.34	0.32	0.25
	DseekR1	0.48	0.47	0.45	0.46	0.44	0.42	0.41	0.47	0.38	0.37	0.36	0.40
NHC	Llama	0.57	0.55	0.56	0.58	0.34	0.32	0.33	0.35	0.53	0.51	0.52	0.54
	Gemma	**0.71**	**0.76**	**0.71**	**0.71**	**0.69**	**0.67**	**0.68**	**0.70**	**0.72**	**0.70**	**0.71**	**0.73**
	Mistral	0.14	0.12	0.13	0.15	0.37	0.35	0.36	0.38	0.29	0.27	0.28	0.30
	Qwen	0.23	0.21	0.22	0.24	0.43	0.41	0.42	0.44	0.43	0.41	0.42	0.44
	Phi	0.67	0.65	0.66	0.67	0.21	0.19	0.20	0.22	0.18	0.16	0.17	0.26
	Dseek7B	0.22	0.20	0.21	0.23	0.21	0.19	0.20	0.22	0.45	0.43	0.44	0.46
	DseekR1	0.65	0.63	0.64	0.66	0.67	0.65	0.66	0.68	0.68	0.66	0.67	0.69

Table 4: Performance comparison of models across five tasks for zero-shot, one-shot, and chain-of-thought (CoT) prompting.

five tasks. This variability underscores the heterogeneous linguistic and semantic demands of multitask Bengali text classification, in which different tasks require distinct types of knowledge and reasoning.

Qwen demonstrates the strongest and most stable performance in Sentiment Analysis (SA), achieving F1 scores of 0.95–0.96 across all prompting strategies. The relative insensitivity to prompting variations suggests that sentiment classification primarily relies on coarse-grained polarity cues that transfer effectively across languages. This indicates that Qwen's multilingual pretraining likely provides sufficient semantic grounding for polarity detection without requiring task-specific adaptation. In contrast, Gemma achieves superior performance on more structurally and semantically complex tasks, including Fake News Detection (FND), News Headline Categorization (NHC), and Text Emotion Analysis (TEA). These tasks demand contextual reasoning, pragmatic interpretation, and fine-grained label discrimination. Gemma's comparatively strong instruction-following alignment appears better suited to handling multi-class classification scenarios that require nuanced semantic differentiation. Llama exhibits competitive zero-shot performance on Aggressive Text Detection (ATD). This observation further supports the notion that task complexity and cultural specificity significantly influence the effectiveness of cross-lingual transfer.

Collectively, these results indicate that model performance in Bengali multitask classification is highly task-dependent and closely tied to the interaction between pretraining distribution, instruction alignment, and the linguistic characteristics of each classification objective.

Does CoT help in these low-resource tasks? Chain-of-thought prompting does not consistently improve performance and frequently degrades it. Although CoT is theoretically beneficial for tasks requiring structured reasoning (Wei et al., 2022), its effectiveness depends on the availability of reliable internal knowledge representations. In Bengali, where pretraining exposure is comparatively limited, CoT often produces fluent but semantically unreliable reasoning chains. For example, models optimized for reasoning exhibit significant degradation under CoT on certain tasks. Instead of correcting errors, the additional reasoning steps amplify

uncertainty. These findings suggest that CoT requires sufficiently rich language-specific priors to be effective. In low-resource contexts, it may increase the risk of hallucinations by encouraging overconfident yet weakly grounded intermediate reasoning steps (Ahmed et al., 2025).

How do the models perform after PEFT? Table 5 reports the performance of instruction-tuned models fine-tuned with QLoRA (PEFT), and the improvements over the best ICL baselines are consistent and often substantial. The most dramatic gains are observed on the tasks that proved hardest under ICL. For TEA, the best ICL F1 (Gemma, zero-shot) was 0.55; after PEFT fine-tuning, Gemma-3-4B achieves an F1 of 0.74 — a gain of 19 percentage points. Similarly, for NHC, the best ICL F1 score was 0.71 (Gemma, zero-shot), whereas PEFT raises it to 0.83. These results confirm that for culturally embedded tasks such as emotion analysis and news categorization in Bengali, task-specific adaptation to in-domain data is far more effective than any prompting strategy. For ATD, PEFT enables Mistral-7B to reach an F1 of 0.93, compared to a best ICL F1 of 0.82. On FND, Gemma-3-4B achieves an F1 of 0.97 — more than 9 points above its own best ICL performance. Across all five tasks, Gemma-3-4B consistently ranks among the top-performing PEFT models, achieving the highest F1 on SA (0.97), FND (0.97), TEA (0.74), and NHC (0.83). Mistral-7B leads only on ATD (F1: 0.93) but exhibits a catastrophic failure on FND after fine-tuning (F1: 0.06), suggesting a label prediction collapse that warrants further investigation. The consistency of PEFT gains across models and tasks supports the conclusion that QLoRA fine-tuning efficiently surfaces latent multilingual capacity that prompting alone cannot elicit. By adapting the model's lower-rank projection matrices to Bengali-specific lexical and morphological distributions, fine-tuning allows even relatively compact models (3B–7B parameters) to substantially close the gap with larger systems evaluated in zero-shot settings.

Which task is most challenging for the models? Across both ICL and PEFT settings, a clear difficulty ordering emerges among the five tasks. SA is the least challenging, with top PEFT models achieving an F1 score above 0.97, whereas TEA remains the hardest, achieving a maximum F1 of 0.74 even after fine-tuning. This ordering reflects the varying degree to which task-relevant knowledge can be transferred from high-resource pretraining distributions. Sentiment polarity is a relatively universal semantic property; emotion, hate, and argumentation in Bengali, by contrast, are deeply tied to cultural register, code-switching practices, and community-specific linguistic conventions that multilingual LLMs are unlikely to have internalized from pretraining data alone.

Which model emerges as the most effective backbone for multitask Bengali text classification across tasks? Among the evaluated models, Gemma-3-4B emerges as the most suitable backbone for multitask Bengali text classification, demonstrating the most consistent cross-task performance under both in-context learning and parameter-efficient fine-tuning.

Why does DeepSeek-R1 perform poorly despite being a larger model? Despite being among the larger models evaluated, R1 underperforms relative to its scale on several tasks. In particular, its CoT performance is consistently inferior to its own zero-shot performance — most dramatically on ATD, where CoT reduces its F1 from 0.77 to 0.42. This behavior is a direct consequence of R1's training objective, which optimizes for systematic chain-of-thought reasoning. When applied to Bengali classification tasks, this inductive bias is counterproductive: the model is compelled to reason through a language and domain it has limited knowledge of, producing confident but incorrect reasoning chains. This finding highlights an important limitation of reasoning-optimized LLMs in low-resource languages.

Task	Model	Pr	Re	F1	Acc
SA	Llama-3.2-4B	0.83	0.85	0.84	0.85
	Gemma-3-4B	0.96	0.97	0.97	0.97
	Mistral-7B-it	0.89	0.92	0.90	0.91
	Qwen-2.5-8B-it	0.94	0.94	0.94	0.94
	Phi-3	0.84	0.85	0.84	0.86
ATD	Llama-3.2-4B	0.89	0.89	0.89	0.89
	Gemma-3-4B	0.91	0.91	0.91	0.91
	Mistral-7B-it	0.93	0.93	**0.93**	0.93
	Qwen-2.5-8B-it	0.89	0.89	0.89	0.89
	Phi-3	0.91	0.91	0.91	0.91
FND	Llama-3.2-4B	0.89	0.89	0.89	0.89
	Gemma-3-4B	0.97	0.97	**0.97**	0.97
	Mistral-7B-it	0.67	0.16	0.06	0.16
	Qwen-2.5-8B-it	0.83	0.84	0.83	0.85
	Phi-3	0.93	0.73	0.81	0.73
TEA	Llama-3.2-4B	0.63	0.61	0.62	0.63
	Gemma-3-4B	0.74	0.74	**0.74**	0.74
	Mistral-7B-it	0.73	0.70	0.69	0.70
	Qwen-2.5-8B-it	0.67	0.67	0.67	0.67
	Phi-3	0.65	0.66	0.65	0.66
NHC	Llama-3.2-4B	0.76	0.77	0.76	0.77
	Gemma-3-4B	0.83	0.83	**0.83**	0.83
	Mistral-7B-it	0.81	0.81	0.81	0.81
	Qwen-2.5-8B-it	0.77	0.77	0.77	0.77
	Phi-3	0.27	0.40	0.30	0.40

Table 5: Performance comparison of different instruction-tuned models across five tasks.

5.1. Ablation Study

We conducted an ablation study on LoRA rank, LoRA α, and LoRA dropout for PEFT. Due to the high computational cost of training Gemma-3 4B Instruct, these experiments were limited to the ATD dataset only. Multiple combinations of LoRA rank, α, and dropout were evaluated, as summarized in Table 6.

Based on the ablation results, Set-8 from Table 6 was selected as the final configuration to develop the best-performing Gemma-3 4B model. Three key factors guided this decision. First, Set-8 achieved comparable or slightly better validation performance than higher-rank configurations (e.g., rank-32) and exhibited more stable generalization across validation splits. Second, it required significantly fewer computational resources than higher ranks (32, 64), including reduced GPU memory usage and faster training times, making it more practical for large-scale fine-tuning. Finally, increasing the LoRA rank beyond this setting did not yield consistent performance gains, indicating diminishing returns relative to the added computational cost. Therefore, Set-8 was chosen as the optimal trade-off between performance, generalization, and efficiency, and it was used for all subsequent experiments and analyses with all other models.

Set	LoRA r	LoRA α	Dropout	Pr	Acc	F1
1	32	64	0	0.92	0.91	0.91
2	8	8	0	0.91	0.90	0.90
3	8	32	0.03	0.89	0.88	0.89
4	16	128	0	0.91	0.91	0.91
5	8	8	0	0.87	0.84	0.84
6	16	64	0	0.91	0.91	0.91
7	32	32	0	0.91	0.91	0.91
8	16	32	0	0.91	0.91	0.91
9	16	16	0.01	0.91	0.90	0.90
10	64	64	0.05	0.91	0.87	0.87

Table 6: Ablation Study on LoRA Hyperparameters (r, α, Dropout) in ATD

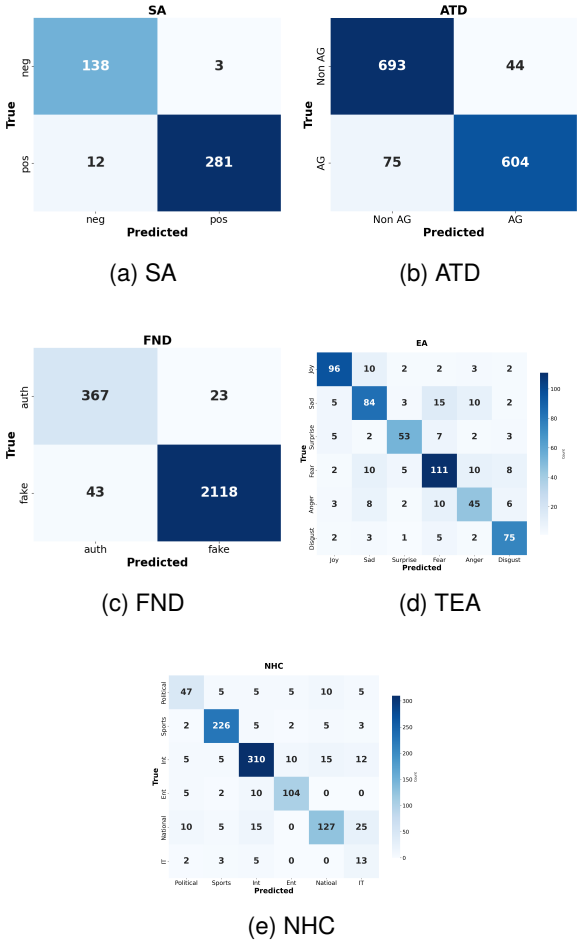
5.2. Error Analysis

As shown in the previous section, the instruction-tuned Gemma-3 4B model exhibits the best overall performance for Bengali text classification.

5.2.1. Quantitative Error Analysis

Figure 3 presents the confusion matrices of the proposed Gemma-3 4B model for five tasks.

For the SA task, the model achieved 138 true negatives (TN) and 281 true positives (TP), indicating strong performance. For the ATD task, it recorded 693 TN and 694 TP; for the FND task, it reached 367 TN and 2118 TP. These results demonstrate excellent performance across all three binary classification tasks. However, performance declined on the EC and NHC tasks with



(a) SA (b) ATD

(c) FND (d) TEA

(e) NHC

Figure 3: Confusion matrices of all five tasks for the proposed model.

six classes, which made them more complex. In these tasks, the Gemma-3 4B model exhibited a noticeable decline in true positives and true negatives compared with its binary classification performance, highlighting its limitations in making fine-grained multiclass distinctions.

5.2.2. Qualitative Error Analysis

Table 7 presents examples of both correctly and incorrectly classified predictions across the five tasks.

These predictions are generated by the Gemma-3 4B instruction-tuned version. In the first three samples, the model correctly identifies the class, demonstrating its effectiveness in clear, less-ambiguous cases. However, in the last two examples, the model fails to predict the correct label. The fourth sample, taken from the news headline categorization task, contains an inherently ambiguous headline that could reasonably belong to multiple categories. This ambiguity likely contributed to the model's misclassification. In the final example, from the emotion analysis task, the distinction between the predicted and actual class is subtle—

Text Samples	Actual	Predicted
একটা প্রকাশনা কিভাবে এমন একটা বই প্রকাশ করে। (How does a publisher publish such a book.)	Negative	Negative
হাইরে মানুষ নিজে তৈরি করে আবার নিজে পূজা করে। (Oh human! You create it yourself, and then worship it yourself.)	Aggressive	Aggressive
ইসলামী মতিবেদকখতনা করেও রেহাই পেলেন না বিশিষ্ঠ কবি, হেকিমী চিকিতসক ও নও মুসলমান ফরহাদ মজহার লুংগি। (Even after adopting Islamic ideology and circumcision, the distinguished poet, Hakimi physician, and new Muslim Farhad Mazhar did not escape criticism.)	Fake	Fake
তাদের মিটিং-মিছিলের জন্য দেশ স্বাধীন করা হয়নি: হানিফ। (The country wasn't liberated for their rallies and protests: Hanif.)	Politics	National
রিয়াদের নিজের উপরই রাগ উঠে কি দরকার ছিল ঘাবেড় দেওয়ার সত্য কথাটুকু বলেই হতো। (Riyad got angry at himself — what was the need to scare them? He could've just told the truth.)	Anger	Disgust

Table 7: Some correctly and incorrectly classified samples by Gemma-3-4B.

even for humans. The text's emotional tone is nuanced, making it especially challenging for the model to interpret accurately. This highlights a common difficulty in fine-grained emotion classification: overlapping emotional cues can lead to confusion.

6. Conclusion

This paper presents a comprehensive evaluation of several LLMs, focusing on their instruction-tuned versions. Gemma-3 4B consistently outperformed other large language models, including Llama, Mistral, Qwen, and Phi, across most tasks, despite its relatively small size. Notably, the instruction-tuned Mistral-7B-it model achieved the best results in aggressive text detection. This study also compared prompting techniques, finding that instruction tuning generally improves performance, and that zero-shot prompting often yields better results than one-shot or chain-of-thought methods. Future work will focus on further extending evaluations to a broader range of domains and investigating more advanced instruction-tuning strategies. Additionally, the latest models, such as GPT-5 and Gemini-2.5 Flash, can be evaluated to assess their effectiveness in handling these tasks under in-context learning settings.

Limitations

Although the proposed method demonstrates satisfactory performance across various text classification tasks, several significant limitations remain unaddressed.

- The instruction-tuned Gemma-3 4B model achieves high accuracy on binary classification tasks; however, its performance declines on complex multiclass classification tasks.
- The model was adapted using Quantized Low-Rank Adaptation (QLoRA) rather than full fine-tuning, which may have potentially restricted task alignment and overall classification accuracy.
- Prompting strategies, including zero-shot, one-shot, and chain-of-thought approaches, yield inconsistent results across different tasks and model architectures.
- Several instruction-tuned models lack explicit training on Bengali-language data, which limits their ability to capture Bengali-specific linguistic features.
- This study relies on monolingual Bengali datasets and excludes Bangla-English code-mixed data, which is common in real-world usage.
- The research scope is restricted to classification tasks and does not investigate additional natural language processing (NLP) tasks, including text generation, summarization, or question answering.

Acknowledgment

This work was supported by the Directorate of Research & Extension (DRE), Chittagong University of Engineering & Technology (CUET), Chittagong, Bangladesh, under the Grant Number CUET/DRE/2023-2024/CSE/025.

Ethical Considerations

This study used only publicly available, pre-existing datasets to evaluate five text classification tasks. No new human data were collected or annotated. All datasets were released for research under appropriate licenses, and all sources are correctly cited. Some tasks may contain sensitive or harmful language. To mitigate this risk, the datasets were used exclusively within their original research context, and dangerous content was

not reproduced or promoted beyond what was necessary for model evaluation. All experiments adhered to responsible artificial intelligence research practices, emphasizing transparency, reproducibility, and fairness while minimizing potential misuse.

Data/Code Availability Statement

The datasets and source code used in this study will be available at: https://github.com/CUET-NLP-Lab/bengali-llm-multitask-classification

Bibliographical References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Shamima Afroz, Kawsar Ahmed, and Mohammed Moshiul Hoque. Leveraging multi-task learning for detecting aggression, emotion, violence, and sentiment in bengali texts. In *5th Muslims in ML Workshop co-located with NeurIPS 2025*.

Kawsar Ahmed, Md Osama, Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2025. Bennumeval: A benchmark to assess llms' numerical reasoning capabilities in bengali. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17782–17799.

Shawly Ahsan, Fairooz Tasnia, Nafisa Tabassum, Avishek Das, Mohammed Moshiul Hoque, and Nazmul Siddique. 2023. Classifying textual sentiment using bidirectional encoder representations from transformers. In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Aysha Akther, Kazi Masudul Alam, and Rameswar Debnath. 2025. Automatic detection of manipulated bangla news: A new knowledge-driven approach. *Natural Language Processing Journal*, 11:100155.

Neel Alex, Eli Lifland, Lewis Tunstall, Abhishek Thakur, Pegah Maham, C Jess Riedel, Emmie Hine, Carolyn Ashurst, Paul Sedille, Alexis Carlier, et al. 2021. Raft: A real-world few-shot text classification benchmark. *arXiv preprint arXiv:2109.14076*.

Sumaiya Rahman Aodhora and Mohammed Moshiul Hoque. 2024. Tetec: Technical text classification in bengali using ensemble of transformers. In *2024 International Conference on Recent Progresses in Science, Engineering and Technology (ICRPSET)*, pages 1–6.

Adrita Barua, Omar Sharif, and Mohammed Moshiul Hoque. 2021. Multi-class sports news categorization using machine learning techniques: resource creation and evaluation. *Procedia Computer Science*, 193:112–121.

Anirban Bhowmick and Abhik Jana. 2021. Sentiment analysis for bengali using transformer based models. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 481–486.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Avishek Das, Mohammed Moshiul Hoque, Omar Sharif, M Ali Akber Dewan, and Nazmul Siddique. 2023. Temox: Classification of textual emotion using ensemble of transformers. *IEEE Access*, 11:109803–109818.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Fuad Ibne Jashim Farhad, Shah Imran, Md Mehedi Hasan Santo, Mahbub Khan, Anamul Sakib, Md Shahidur Rahman, Md Ariful Islam, Rezaul Haque, and Shafiur Rahman. 2024. Addressing misinformation in bengali media: A hybrid deep learning solution. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 774–779. IEEE.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024.

The llama 3 herd of models. *arXiv preprint arXiv:2407. 21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Md Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023. Zero-and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv preprint arXiv:2308.10783*.

Md. Ali Hider, Shawly Ahsan, Jawad Hossain, and Mohammed Moshiul Hoque. 2024. Emotion classification in bengali-english code-mixed data using transformers. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 3529–3535.

Eftekhar Hossain. 2023. Bangla news headlines categorization. GitHub repository. Accessed: Jun. 27, 2025.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2021a. Sentiment polarity detection on bengali book reviews using multinomial naive bayes. In *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2020*, pages 281–292. Springer.

Md. Rajib Hossain, Mohammed Moshiul Hoque, M. Ali Akber Dewan, Enamul Hoque, and Nazmul Siddique. 2025. Authornet: Leveraging attention-based early fusion of transformers for low-resource authorship attribution. *Expert Systems with Applications*, 262:125643.

Md. Rajib Hossain, Mohammed Moshiul Hoque, M. Ali Akber Dewan, Nazmul Siddique, Md. Nazmul Islam, and Iqbal H. Sarker. 2021b. Authorship classification in a resource constraint language using convolutional neural networks. *IEEE Access*, 9:100319–100338.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. Banfakenews: A dataset for detecting fake news in bangla. *arXiv preprint arXiv:2004.08789*.

Mohsinul Kabir, Md Tahmid Rahman Laskar, Mir Tafseer Nayeem, M Saiful Bari, and Enamul Hoque. 2023. Benllmeval: A comprehensive evaluation into the potentials and pitfalls of large language models on bengali nlp. *arXiv preprint arXiv:2309.13173*.

Prashant Kapil and Asif Ekbal. 2025. A transformer based multi task learning approach to multimodal hate speech detection. *Natural Language Processing Journal*, 11:100133.

Lefteris Loukas, Ilias Stogiannidis, Prodromos Malakasiotis, and Stavros Vassos. 2023. Breaking the bank with chatgpt: few-shot text classification for finance. *arXiv preprint arXiv:2308.14634*.

Mounika Marreddy, Subba Reddy Oota, Lakshmi Sireesha Vakada, Venkata Charan Chinni, and Radhika Mamidi. 2022. Multi-task text classification using graph convolutional networks for large-scale low resource language. In *2022 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Zabir Al Nazi, Md. Rajib Hossain, and Faisal Al Mamun. 2025. Evaluation of open and closed-source llms for low-resource language with zero-shot, few-shot, and chain-of-thought prompting. *Natural Language Processing Journal*, 10:100124.

Tanjela Rahman Rosni, Mahamudul Hasan, Tanni Mittra, Md Sawkat Ali, and Md Hasanul Ferdaus. 2024. Aggressive bangla text detection using machine learning and deep learning algorithms. In *International Conference on Computation of Artificial Intelligence & Machine Learning*, pages 174–183. Springer.

Timo Schick and Hinrich Schütze. 2022. True few-shot learning with prompts—a real-world perspective. *Transactions of the Association for Computational Linguistics*, 10:716–731.

Omar Sharif and Mohammed Moshiul Hoque. 2022. Tackling cyber-aggression: Identification and fine-grained categorization of aggressive texts on social media using weighted ensemble of transformers. *Neurocomputing*, 490:462–481.

Omar Sharif, Mohammed Moshiul Hoque, A. S. M. Kayes, Raza Nowrozy, and Iqbal H. Sarker. 2020. Detecting suspicious texts using machine learning techniques. *Applied Sciences*, 10(18).

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2022. M-bad: A multilabel dataset for detecting aggressive texts and their targets. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 75–85.

Gopendra Vikram Singh, Mauajama Firdaus, Dushyant Singh Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. 2024. Zero-shot multitask

intent and emotion prediction from multimodal data: A benchmark study. *Neurocomputing*, 569:127128.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Hongbin Wang, Chun Ren, and Zhengtao Yu. 2025. Multimodal sentiment analysis based on multiple attention. *Engineering Applications of Artificial Intelligence*, 140:109731.

Zhiqiang Wang, Yiran Pang, and Yanbin Lin. 2024. Smart expert system: Large language models as text classifiers. *arXiv preprint arXiv:2405.10523*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jian Zhang, Ke Yan, and Yuchang Mo. 2021. Multi-task learning for sentiment analysis with hard-sharing and task recognition mechanisms. *Information*, 12(5):207.

A. Prompt Examples

In this task, we have explored three types of prompting. Tables 8, 9, and 10 illustrate all types of prompt templates used in this research. In Table 9, one-shot samples are provided. However, in this Table, we show only an example for one class. But while experimenting, we provided one example for each class.

Task	Prompt
SA	Please classify the sentiment of this review. The answer should be either `1` (positive) or `0` (negative), based on the sentiment expressed. # Review: `<review>`
ATD	Please classify whether this Bengali sentence is Aggressive or non-aggressive. The answer should be either `1` (Aggressive) or `0` (Non-Aggressive), based on the sentence. # Sentence: `<sentence>`
FND	Please determine whether this Bengali news is genuine or not. The answer should be either `1` (Fake) or `0` (Authentic), based on the news. # News: `<news>`
NHC	Classify the following news headline into one of the predefined categories. Use the corresponding label number: `politics: 0, sports: 1, international: 2, entertainment: 3, national: 4, IT: 5.` # Headline: `<headline>` # Answer: `[num]`
TEA	Classify the emotion expressed in the following Bengali text into one of the predefined categories. Use the corresponding label number: `0: Joy, 1: Sadness, 2: Surprise, 3: Disgust, 4: Anger, 5: Fear.` # Sentence: `<sentence>` # Answer: `[label]`

Table 8: Prompt templates for zero-shot prompt across all the tasks

Task	Prompt
SA	Please classify the sentiment of this review. The answer should be either `1` (positive) or `0` (negative). **Example:** # Review: এই সিনেমাটি দারুণ লেগেছে # Answer: `1` **Now classify:** # Review: `<review>`
ATD	Please classify whether this Bengali sentence is Aggressive or Non-Aggressive. The answer should be either `1` (Aggressive) or `0` (Non-Aggressive). **Example:** # Sentence: আমি তোমাকে মেরে ফেলবো # Answer: `1` **Now classify:** # Sentence: `<sentence>`
FND	Please classify whether this Bengali news is Fake or Authentic. The answer should be either `1` (Fake) or `0` (Authentic). **Example:** # News: আগামীকাল চাঁদে ফুটবল খেলা হবে # Answer: `1` **Now classify:** # News: `<news>`
NHC	Classify the following news headline into one of the predefined categories. Use the corresponding label number: `politics: 0, sports: 1, international: 2, entertainment: 3, national: 4, IT: 5.` **Example:** # Headline: বাংলাদেশ ক্রিকেট দল জিতেছে # Answer: `1` **Now classify:** # Headline: `<headline>`
TEA	Classify the emotion expressed in the following Bengali text into one of the predefined categories. Use the corresponding label number: `0: Joy, 1: Sadness, 2: Surprise, 3: Disgust, 4: Anger, 5: Fear.` **Example:** # Sentence: আমি আজ খুব খুশি # Answer: `0` **Now classify:** # Sentence: `<sentence>`

Table 9: Prompt templates for one-shot prompt across all the tasks

Task	Prompt
SA	**Task:** Classify the sentiment of the following Bangla review as `1` (positive) or `0` (negative). **Steps:** 1. **Analyze Content:** Break the review into phrases or sentences. 2. **Identify Sentiment Indicators:** Look for positive (e.g., "অসাধারণ'') or negative (e.g., ``খারাপ'') words/phrases. 3. Evaluate Context: Consider how words are used, including sarcasm or mixed sentiment. 4. Determine Tone: Assess the overall tone based on key phrases. 5. Classify: Assign `1` for positive and `0` for negative. Now classify: # Review: `<review>`
ATD	**Task:** Classify whether the following Bangla sentence is Aggressive (`1`) or Non-Aggressive (`0`). **Steps:** 1. **Analyze Content:** Break the sentence into meaningful parts. 2. **Identify Aggression Indicators:** Look for threatening or harmful expressions (e.g., "মেরে ফেলবো''). 3. Evaluate Intensity: Check the severity of the words used and their target. 4. Determine Tone: Assess whether the tone is hostile or neutral. 5. Classify: Assign `1` for Aggressive and `0` for Non-Aggressive. Now classify: # Sentence: `<sentence>`
FND	**Task:** Classify whether the following Bangla news is Fake (`1`) or Authentic (`0`). **Steps:** 1. **Analyze Content:** Break the news statement into key claims. 2. **Check Plausibility:** Identify whether the claim sounds realistic or exaggerated. 3. **Identify Unrealistic Elements:** Look for impossible or illogical events (e.g., "চাঁদে ফুটবল খেলা হবে''). 4. Evaluate Source-Like Tone: Assess if the sentence resembles factual reporting or satire. 5. Classify: Assign `1` for Fake and `0` for Authentic. Now classify: # News: `<news>`
NHC	**Task:** Classify the following Bangla news headline into one of the predefined categories: `politics: 0, sports: 1, international: 2, entertainment: 3, national: 4, IT: 5`. **Steps:** 1. **Analyze Content:** Break the headline into key subjects and actions. 2. **Identify Keywords:** Detect topic-related words (e.g., "ক্রিকেট'' → sports, ``সরকার'' → politics). 3. Match with Categories: Compare keywords with the predefined category list. 4. Resolve Ambiguity: If multiple categories fit, select the most dominant one. 5. Classify: Assign the corresponding label number. Now classify: # Headline: `<headline>`
TEA	**Task:** Classify the emotion expressed in the following Bangla text into one of the predefined categories: `0: Joy, 1: Sadness, 2: Surprise, 3: Disgust, 4: Anger, 5: Fear`. **Steps:** 1. **Analyze Content:** Break the sentence into emotion-bearing parts. 2. **Identify Emotion Indicators:** Look for explicit words or expressions (e.g., "খুশি'' → Joy, ``রাগ'' → Anger). 3. Evaluate Context: Consider implied emotions or indirect expressions. 4. Determine Dominant Emotion: Choose the strongest emotion if multiple exist. 5. Classify: Assign the corresponding label number. Now classify: # Sentence: `<sentence>`

Table 10: Prompt templates for zero-shot CoT across all tasks with reasoning steps

Nwāchā Munā: A Devanagari Speech Corpus and Proximal Transfer Benchmark for Nepal Bhasha ASR

Rishikesh Kumar Sharma, Safal Narshing Shrestha, Jenny Poudel, Rupak Tiwari, Arju Shrestha, Rupak Raj Ghimire, Bal Krishna Bal
Information and Language Processing Research Lab, Kathmandu University
{rishi70612, safalnarsingh, jennypoudel100, rupaktiwari18, arzzustha}@gmail.com,
rughimire@gmail.com, bal@ku.edu.np

Abstract

Nepal Bhasha (Newari), an endangered language of the Kathmandu Valley, remains digitally marginalized due to the severe scarcity of annotated speech resources. In this work, we introduce **Nwāchā Munā** (/nwaːtʃaː munaː/), a newly curated 5.39-hour manually transcribed Devanagari speech corpus for Nepal Bhasha, and establish the first benchmark using script-preserving acoustic modeling. We investigate whether proximal cross-lingual transfer from a geographically and linguistically adjacent language (Nepali) can rival large-scale multilingual pretraining in an ultra-low-resource Automatic Speech Recognition (ASR) setting. Fine-tuning a Nepali Conformer model reduces the Character Error Rate (CER) from a 52.54% zero-shot baseline to 17.59% with data augmentation, effectively matching the performance of the multilingual *Whisper-Small* model despite utilizing significantly fewer parameters. Our findings demonstrate that proximal transfer from Nepali language serves as a computationally efficient alternative to massive multilingual models. We openly release the dataset and benchmarks to digitally enable the Newari community and foster further research in Nepal Bhasha.

Keywords: Automatic Speech Recognition, Nepal Bhasha, Speech Corpus, Low-Resource Speech Recognition

1. Introduction

In the current space of AI, high-resource languages dominate in terms of resource availability while indigenous languages remain digitally marginalized. In a country like Nepal, with 124 official languages, this divide is especially vivid, creating a need for technologies like ASR to bridge the gap. Nepal Bhasha (Newari) serves as a prime example of this imbalance, despite being actively spoken by over 860,000 people in Nepal. Genetically, Nepal Bhasha belongs to the Tibeto-Burman branch of the Sino-Tibetan family (Eberhard et al., 2026). Even with a six-century-old history and a designation as one of the official working languages of Bagmati Province (Post Report, 2024), Nepal Bhasha is currently classified by UNESCO as a "definitely endangered" language (Moseley, 2010). Building a system such as ASR empowers these communities to maintain their linguistic heritage while fully participating in the modern technological landscape.

The modern field of speech recognition has transitioned from fragmented components based on Hidden Markov Models (Rabiner, 1989) to unified end-to-end (E2E) architectures such as Transformer (Vaswani et al., 2017) and Conformer (Gulati et al., 2020). These modern architectures require large datasets to achieve better performance. Recent advancements have yielded robust ASR systems for languages like Nepali and Hindi, but indigenous languages such as Nepal Bhasha remain severely bottlenecked by an extreme scarcity of annotated speech data. This research addresses the fundamental gap by developing a speech corpus for the Newari language—Nwāchā Munā. To further demonstrate the usefulness of this dataset, we establish an initial model benchmark and synthesize modern ASR architectures with cross-lingual transfer learning, leveraging a proximate source language to overcome data constraints.

From this research, we aim to provide a scalable blueprint for other underrepresented languages in the region.

Our contributions are threefold:

1. We release Nwāchā Munā, a carefully curated 5.39-hour Devanagari speech corpus for Nepal Bhasha.
2. We provide the first controlled comparison between proximal transfer (Nepali $\rightarrow$ Newari) and multilingual pretraining (Whisper) in this ultra-low-resource setting.
3. We show that script-preserving proximal transfer can match large multilingual models while requiring substantially fewer parameters and computational resources.

The remainder of this paper is structured as follows: Section 2 reviews related works in low-resource ASR. Section 3 describes the dataset

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 105–114
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

and methodology including data collection and model training strategies. Section 4 presents the experimental results, discussions and error analysis. Section 5 presents the conclusion followed by ethical statement and limitations.

2. Related works

One major obstacle in building ASR systems for underrepresented languages is the lack of sufficient speech data. Older methods like Hidden Markov Models required less data but struggled to capture intricate acoustic patterns. With time, new architectures like Transformers (Vaswani et al., 2017) and Conformer (Gulati et al., 2020) have outperformed older statistical methods, but they require larger datasets. To address this issue, researchers have increasingly turned to transfer learning. For instance, Cheng et al. (2024) demonstrated clear ASR performance gains for two at-risk Austronesian languages, Amis and Seediq, despite having minimal available speech samples. They achieved this by utilizing a novel data-selection scheme to automatically extract phonetically similar utterances from a broad multilingual corpus to augment their training data. Likewise, significant progress has been made specifically within the South Asian region. The AI4Bharat initiative (Javed et al., 2022) advanced regional ASR by curating over 17,000 hours of raw speech data across 40 Indian languages for continued pre-training.

In the context of Nepali ASR, early efforts relied on Hidden Markov Models (HMM) (Ssarma et al., 2017) for isolated word recognition, which struggled with continuous speech. The transition to deep learning began with Regmi et al. (2019), they implemented Recurrent Neural Networks (RNN) with Connectionist Temporal Classification (CTC) loss. This was further refined by Bhatta et al. (2020) with a hybrid CNN-GRU-CTC model. Furthermore, Regmi and Bal (2021) implemented end-to-end speech recognition approach for the Nepali language. To address training instability in deeper networks, Dhakal et al. (2022) proposed a CNN-BiLSTM-ResNet architecture, achieving a CER of 17.06%. Concurrently, research began shifting towards data efficiency and cross-lingual techniques. For example, Bansal et al. (2020) explored an acoustic-phonetic approach, demonstrating that leveraging monolingual and cross-lingual information could improve performance in low-resource settings. Joshi and Shrestha (2023) applied Self-Attention Networks (SAN) to capture long-range dependencies better than recurrent models. Building on these modern architectures, Paudel et al. (2023) successfully applied a combination of CNNs and Transformers for large vocabulary continuous speech recognition.

Current state-of-the-art systems have fully embraced E2E Transformer-based architectures. Poudel et al. (2025) introduced `NepConformer`, a Conformer-based Nepali ASR model achieving a CER of `6.01%`. Recent focus has also turned to data selection, tokenization and resource efficient strategies. Ghimire et al. (2023a) proposed an Active Learning framework to select the most informative samples for speech annotation. Another work (Ghimire et al., 2023b), demonstrated that using linguistically motivated sub-word units (syllables) alongside active sampling significantly outperforms standard methods. Work done by Ghimire et al. (2025) showed that using adapter-based Parameter-Efficient-Fine-Tuning (PEFT) alone improves Word Error Rate (WER) by 19% compared to `Whisper-large-v2`. Extending the utility of efficient adaptation, Pantha et al. (2025) demonstrated the effectiveness of PEFT for speech personalization among Nepali speakers. Most recently, Ghimire et al. (2023b) implemented new complementary loss function for addressing grammatical consistency during model training and fine-tuning. Their Rule-Based Character Constituency Loss (RBCCL) help to reduce Word Error Rates (WER) from 47.1% to 23.41% by penalizing grammatically impossible character sequences during training.

While Nepali has seen consistent growth in the field of ASR, research on Nepal Bhasha remains sparse. A notable exception is the work by Meelen et al. (2024), who presented the first Newari ASR, where they achieved a CER of 12%. Their evaluation is conducted in Romanized transliteration, this fails to preserve Devanagari orthography. Their study highlighted that while transfer learning is effective, the lack of standardized orthography and limited data results in higher WER compared to Nepali. They emphasized the need for ”orthography standardization” and data augmentation techniques like SpecAugment (Park et al., 2019) to make ASR viable for these endangered languages.

Taken together, these findings highlight a persistent gap: prior work lacks ASR systems trained on the Devanagari script using carefully curated low-resource datasets. This research clears these gaps by developing Nwāchā Munā. We utilise Devanagari script for transcribing speech of Nepal Bhasha. We also use data augmentation techniques to train robust ASR models using various cross-lingual transfer strategies and report CER for each technique.

3. Methodology

3.1. Data Collection Technique

The development of the Nepal Bhasha ASR required a carefully designed data collection process to ensure authenticity and reliability. Both textual and acoustic resources were gathered considering language and speaker diversity. The objective was to construct a dataset that reflects natural language usage while maintaining consistency and structural integrity. The following subsections describe the procedures adopted for textual and audio data acquisition.

3.1.1. Textual Data Acquisition

To establish a baseline for the Nepal Bhasha ASR system, a comprehensive textual corpus was compiled from different sources of digital and print media. The data collection strategy included the use of both formal and everyday language to capture the full range of vocabulary. Digital data were extracted from Nepal Bhasha Wikipedia [1] and the OSCAR (Open Super-large Crawled Aggregated coRpus) (Ortiz Suárez et al., 2020) dataset while print-based resources included regional newspapers, literary manuscripts from the Bloom Library [2], and primary school textbooks. In addition to the formal texts obtained, a targeted set of daily conversational sentences was curated manually to incorporate texts that were previously missing. The raw text was rigorously pre-processed where non-target language tokens, numerical digits, and idiosyncratic symbols were manually removed to ensure high fidelity and relevance. Table 1 shows an example of text normalization, where raw text containing Devanagari numerals and symbols is converted into a phonetic, standardized form for acoustic modeling.

Table 1: Example of Text Normalization for Nepal Bhasha ASR

Condition	Sentence Structure
Before	''छगु ब्वसय् २१% ब्व दु।'' chagu bwasay 21% bwa du *The sentence contains 21 as a number.*
After	छगु ब्वसय् नीछगू सयेक ब्व दु chagu bwasay nīchagū sayeka bwa du *The sentence contains 21 in textual form*

The overall text corpus statistics and sentence length distribution are presented in Table 2 and 3:

Table 2: Nwāchā Munā Corpus Overview

Metric	Count
Total Sentences	5,727
Total Words	27,644
Unique Words in Corpus	8,599

Table 3: Sentence Length Distribution in Nwāchā Munā Corpus

Sentence Length	Count
Short (0–5 words)	4,323
Medium (6–10 words)	1,073
Long (> 10 words)	331

3.1.2. Audio Data Acquisition

Primary acoustic data was gathered through a dual-modal strategy involving original field recordings and the transcription of existing web-based audio, ensuring the model's exposure to varied acoustic environments. To maintain technical consistency across the dataset, all audio was standardized to a 16 KHz sampling rate and stored in a mono-channel WAV format, providing the necessary fidelity for deep learning-based feature extraction. For the recording phase, 18 native speakers from Banepa, Dhulikhel, Panauti, and Patan volunteered based on criteria such as native fluency and linguistic knowledge. While Newari exhibits location-based dialectal variations, the use of a standardized text across all recordings ensured that the resulting differences remained primarily tonal. To enhance the model's robustness against speaker-dependent variability, the data was stratified by age and gender. Audio recordings were captured using built-in smartphone microphones in an open environment with minimal acoustic disturbance. This field collection of 4 hours and 21 minutes of speech was further supplemented by approximately one hour of web-sourced audio (Meelen et al., 2024), which was originally in Romanized form and subsequently transliterated into Devanagari script by community members.

The overall characteristics of the Nwāchā Munā speech corpus are summarized in Tables 4-6. Table 4 presents the corpus-level statistics, including the total number of utterances, total duration, and mean utterance length. Table 5 details the age distribution of the same participant set, while Table 6 provides the gender distribution of the 18 speakers contributing to the corpus.

[1] https://new.wikipedia.org/wiki/
[2] https://bloomlibrary.org/language:new

Table 4: Overview of the Nwāchā Munā

Metric	Value
Total Utterances	5,727
Total Duration	5.39 hours
Mean Utterance Length	3.39 sec

Table 5: Age Distribution of Speakers in the Nwāchā Munā Corpus

Age Group	Number of Speakers
16–25	6
26–30	2
30+	10
Total Speakers	18

Table 6: Gender Distribution of Speakers in the Nwāchā Munā Corpus

Gender	Number of Speakers
Male	10
Female	8
Total Speakers	18

3.2. Model Training Strategies

We adopted a comparative experimental framework to systematically evaluate cross-lingual transfer strategies for Newari automatic speech recognition. Our central hypothesis is that acoustic and orthographic proximity between Nepali and Newari enables effective encoder reuse, thereby reducing the need for large multilingual pretraining. To test this hypothesis, we structured our investigation into three core components: 1) acoustic modeling, 2) language modeling and decoding, and 3) semi-supervised learning.

In the domain of acoustic modeling, we evaluated distinct training strategies starting with a zero-shot evaluation of the pre-trained `NepConformer` (Poudel et al., 2025), followed by supervised fine-tuning of NepConformer, and fine-tuning of a massive multilingual model, `Whisper-Small` (244M parameters) on the collected Newari dataset. To test the adaptability of pre-trained acoustic representations, we further distinguished between standard full model fine-tuning and a decoder-only fine-tuning strategy. We also investigated the impact of data augmentation techniques in improving the performance and robustness of the models.

Finally, to address low-resource constraints in decoding, we explored language modeling strategies by incorporating an external KenLM-based n-gram model via shallow fusion to enhance transcription accuracy.

3.2.1. Acoustic Modeling Strategies

The zero-shot evaluation of NepConformer served as a baseline to assess cross-lingual generalization without any exposure to Newari during training. This setup enabled us to measure the extent to which acoustic and linguistic representations learned from Nepali transfer to a closely related but distinct language. Subsequently, we employed transfer learning through supervised fine-tuning on the Newari corpus to adapt the NepConformer model to the target language. This allowed us to measure the gains obtained from domain-specific adaptation compared to zero-shot performance. To test the adaptability of pre-trained acoustic representations, we further distinguished between standard full model fine-tuning and a decoder-only fine-tuning strategy, where the encoder parameters were frozen to assess the sufficiency of source-language acoustic features. Finally, we fine-tuned Whisper-Small to evaluate whether broad cross-lingual pre-training yielded stronger transfer capabilities than a monolingual Nepali trained model.

To combat data scarcity in low-resource Nepal Bhasha ASR, a dual-stage data augmentation strategy was employed. By combining static (offline) and dynamic (online) methods, we introduced diverse acoustic variations to improve model robustness and prevent overfitting. This approach acts as a regularizer, forcing the model to learn invariant phonetic features rather than memorizing the specific acoustic signatures of the limited training set.

3.2.2. Language Modeling and Decoding

To enhance linguistic modeling under low-resource constraints, we used two approaches: decoder-only fine-tuning and shallow fusion with an external language model (KenLM) (Heafield, 2011).

In the decoder-only setup, the encoder parameters were frozen, and only the prediction network and joint network were updated during fine-tuning. By applying the phonetic similarities between Nepali and Newari, this strategy conserves the transferable acoustic representations obtained during pre-training while allowing the decoder to adapt specifically to Newari linguistic patterns.

Independent of decoder adaptation, we also integrated an external n-gram language model trained using KenLM and applied shallow fusion

during inference. This introduces explicit linguistic priors to capture word-level dependencies that the limited paired data might miss. During beam search decoding, the log-probabilities from the acoustic model and the external language model were linearly combined (Toshniwal et al., 2018), with the fusion weight tuned on the development set.

For the baseline system, beam search decoding approximates the most likely sequence:

$$y^* = \arg\max_y \log p(y \mid x)$$

Under shallow fusion, we incorporate the external language model via log-linear interpolation with a word insertion bonus:

$$y^* = \arg\max_y \{\log p(y \mid x) + \alpha \log p_{LM}(y) + \beta |y|_w\}$$

where α controls the language model weight, β is the word insertion bonus, and $|y|_w$ denotes the number of words in the hypothesis y.

Decoder-only fine-tuning improves the model's language understanding by updating only the higher-level components while keeping the encoder fixed, which also reduces computational cost. This setup allows us to observe the effect of adapting the decoder alone in a low-resource setting. While, shallow fusion adds an external language model during decoding without changing the acoustic model itself. By evaluating these two methods independently, we can clearly measure the contributions of internal decoder adaptation and external statistical language modeling to CER, WER, and improve transcription consistency in low-resource settings.

3.2.3. Semi-Supervised Learning

Due to the availability of limited data for Nepal Bhasha, we adopted a semi-supervised pseudo-labeling framework. An unlabeled corpus was collected from publicly available broadcast sources, including radio, podcasts, and television, and segmented into shorter utterances. To mitigate memory issues and alignment degradation in CTC-based training, excessively long utterances were discarded. The remaining audio was transcribed using our best intermediate acoustic model to generate pseudo-labels.

To ensure quality, we applied rigorous filtering. Single-word transcripts were removed, as they were typically noise-induced fragments rather than valid words. We also computed average prediction confidence scores and applied a threshold filter to reduce error reinforcement before incorporating the pseudo-labeled data into our training pipeline.

4. Results and Discussion

4.1. Experimental Setup

The dataset was divided into 80% for training, 10% for validation, and 10% for testing. Due to the limited size of the dataset across 18 participants, speaker representation was maintained in all splits to prioritize overall acoustic diversity rather than enforcing strict speaker independence. Table 7 summarizes the Newari speech corpus split across training, validation, and test sets, showing the number of hours and utterances for each partition.

Table 7: Dataset split for Newari speech corpus

Split	Hours	Utterances
Train	4.31	4575
Validation	0.54	576
Test	0.54	576

For the Conformer model, static augmentation expanded the training set by a factor of 5 via speed perturbation ($0.9\times, 1.1\times$), volume randomization, and noise injection, resulting in 23.05 total hours of audio. Dynamic augmentation was applied with a probability of $p = 0.5$, involving time stretching (rate $0.8\times$–$1.25\times$), pitch shifting (± 4 semitones), and Gaussian noise injection (amplitude 0.001–0.015).

The fine-tuned Nepal Bhasha ASR model utilizes a `Conformer-CTC` architecture with a Byte Pair Encoding (BPE) tokenizer of vocabulary size 128. The 18-layer Conformer encoder has a d-model dimension of 256, 4 attention heads, and a convolution kernel size of 31. Regularization includes a dropout rate of 0.1 across the encoder and attention modules, alongside SpecAugment with 2 frequency masks (width 27) and 2 time masks (width 70). Optimization is performed using AdamW with a learning rate of 1×10^{-4} and a Cosine Annealing schedule with 3,000 warmup steps. T4 GPU with 16GB vRAM and 29GB of system RAM was used for training in Kaggle environment with total training time of 3 hours for standard data and 12 hours for augmented data.

The `Whisper-Small` model was fine-tuned for 10 epochs using a sequence-to-sequence objective, with the decoder explicitly forced to the Nepali language token in order to leverage script compatibility. Optimization was performed using AdamW with a cosine learning rate decay (peak 1×10^{-5}, 10% warmup) and an effective batch size of 16, achieved via gradient accumulation under FP16 mixed-precision. To further ensure stability, a dropout rate of 0.2 to all attention and hidden layers were applied. A T4-GPU with 15GB vRAM was

used for training this model in Google Colab environment with total training time of 4.5 hours.

For language modeling and decoding, we performed a grid search on the development set to identify optimal decoding hyperparameters across all models. We explored different configurations of language model weight (α), word insertion bonus (β), and beam width. Based on this systematic evaluation, a 5-gram KenLM with $\alpha = 0.4$, $\beta = 1.5$, and a beam width of 128 was selected, as it consistently yielded the best performance improvements. The language model was trained on a text corpus consisting of approximately 51k utterances and 478k word tokens, sourced from Nepal Bhasha Wikipedia. Notably, this dataset was reserved solely for language modeling and is distinct from the audio transcription data.

For the semi-supervised learning experiments, an initial unlabeled corpus of 13.65 hours was collected. During the filtering phase, utterances longer than 15 seconds were discarded. After transcribing the audio with our intermediate acoustic model, a 70% confidence threshold was applied to the predictions to ensure pseudo-label quality. Following this rigorous filtering process, 9.33 hours of high-quality pseudo-labeled data remained for experimental training.

4.2. Findings and Analysis

The performance of the models was evaluated using CER as the primary metric. The experiments were conducted on the Nwāchā Munā dataset, while additional evaluations incorporated augmented data and unlabelled speech to examine the effect of data expansion and semi-supervised training. The quantitative results are summarized in Table 8, where only the best-performing models at their optimal training epochs are reported.

Due to differences in script representation, normalization, evaluation protocol and the unavailability of their exact test split, a direct numerical comparison with (Meelen et al., 2024) is not feasible. Instead, our experimental results demonstrate that transferring representations from a neighboring language with a similar script is a viable alternative to massive multilingual pre-training for low-resource ASR, suggesting that linguistic proximity can potentially outweigh model scale in ultra-low-resource South Asian scenarios.

Table 9 shows the orthographic and phonological comparison of the Nepali and Newari Language. Table 10 presents a comparison of model size and computational resource requirements, highlighting that `NepConformer` is significantly more lightweight, with only 30.54 million parameters and lower VRAM usage (8.56 GB) compared to `Whisper-Small`, which requires substantially higher usage (12.38 GB).

Despite possessing significantly fewer parameters and lacking the huge data backing available in multilingual models like Whisper, the `NepConformer` model achieved a baseline CER of 18.72%, effectively matching the performance of the fine-tuned `Whisper-Small` model (18.76%). This parity suggests that the acoustic and orthographic overlap between Nepali and Newari facilitates robust feature extraction, minimizing the need for the extensive capacity of large-scale models. Furthermore, the application of data augmentation proved critical in this data-scarce regime, yielding the state-of-the-art performance of 17.59% CER for `NepConformer` and 17.88% for `Whisper-Small`, highlighting that augmenting scarce audio data can effectively overcome dataset limitations and drive substantial performance gains.

Conversely, a closer examination of our training configurations reveals critical insights into the

Table 8: Summary of Results for Evaluated Modeling Strategies

Strategies (model details)	**Dataset**	**CER (%)**	**WER (%)**
Zero-Shot NepConformer	-	52.54	98.68
Semi-supervised NepConformer	5.39 hours labelled + 9.33 hours Pseudo labeled	19.83	58.28
NepConformer + 5-gram KenLM (Shallow fusion, $\lambda = 0.4$, $\beta = 1.5$, beam 128)	5.39 hours labelled	19.75	**48.46**
Decoder-only NepConformer (Encoder frozen)	5.39 hours labelled	18.77	53.47
Whisper-Small (Fine-tuned on base data)	5.39 hours labelled	18.76	51.44
NepConformer (Fine-tuned on base data)	5.39 hours labelled	18.72	54.34
Whisper-Small + Augmentation (Time stretch, pitch shift, Gaussian noise)	5.39 hours labelled	17.88	48.80
NepConformer + Augmented Data (speed perturbation ($0.9\times, 1.1\times$), volume randomization, and noise injection)	23.05 hours labelled	**17.59**	52.61

Table 9: Orthographic and Phonological Comparison of Nepali and Newari

Features	Nepali	Newari
Script	Devanagari	Devanagari
Nasalization Markers	Yes	Yes (more enhanced)
Agglutination	Moderate	High
Halant Usage	Standard	High Morphological Load

Table 10: Computational Resource and Parameter Comparison

Model	Parameter (Million)	VRAM Used (GB)
NepConformer	30.54	8.56
Whisper(small)	244	12.38

adaptation process. The poor zero-shot performance (52.54% CER) confirms that despite script sharing, the phonological distinctiveness of Newari requires explicit fine-tuning. The *decoder-only* fine-tuning of `NepConformer` yielded results (18.77% CER) nearly identical to full model fine-tuning, indicating that the pre-trained Nepali encoder features are sufficiently generalized to encode Newari speech without modification.

Additionally, the integration of the KenLM n-gram language model was evaluated across all models during decoding. The best results were observed with the NepConformer fine-tuned model, where WER was reduced by approximately 11.7% relative to the baseline. At the same time, the CER increased slightly by 1.37%, as the language model tends to favor standard spellings, which can override phonetically correct but unusual or local forms in Newari speech. This happens partly because the text used to train the language model doesn't always match how people actually speak. Overall, this shows the trade-off between making words more consistent and keeping the natural variations of this low-resource, agglutinative language.

Finally, in the semi-supervised learning approach, incorporating the pseudo-labeled data with the original labeled training set degraded performance, increasing the CER to 19.83% compared to the 17.59% baseline. A qualitative analysis of a random sample of these pseudo-labels revealed three primary sources of error: extensive code-mixing (alternating with Nepali or English), frequent overlapping conversational speech, and highly variable acoustic environments (e.g., background music, distant microphones). Ultimately, this decline highlights the critical challenge that the severe domain shift introduced by uncurated data can completely overshadow the benefits of increased training volume. This suggests that strict domain alignment, speaker diarization, and rigorous filtering are significantly more crucial than raw data quantity for endangered-language ASR.

4.3. Error Analysis

Newari's rich morphological structure, characterized by productive affixation and concatenated morphemes, makes character-level analysis particularly informative. Although the model frequently predicts individual characters correctly, it struggles to assemble them into accurate representations. As a result, the CER remains relatively low, while the WER is substantially higher, reflecting the difficulty of modeling agglutinative formations and complex orthographic concatenations. This is also observed in other work (D K et al., 2025).

Qualitative analysis indicates that transcription errors are primarily associated with specific diacritics: हलन्त (◌्) (vowel suppressor), अनुस्वार (◌ं) (nasalization), चंद्रबिंदु (◌ँ) (nasal vowel), and विसर्ग (◌ः) (aspiration). These markers encode essential phonetic distinctions, including nasalization, short stops, and breathy releases; their omission alters the acoustic-orthographic correspondence. As further illustrated in Table 11, these findings underscore the importance of language-specific grapheme modeling and the development of corpora encompassing both formal and conversational usage.

Table 11: Error Frequency Table

Error Type	% of total errors
Word Boundary Errors	35%
Halant/Cluster Errors	25%
Nasalization & Anusvara Confusion	20%
Lexical Substitution	15%
Function Word & Particle Error	5%

The sample qualitative error are presented in Table 12. This distribution shows that the primary challenges arise not from isolated character misrecognition but from morphological segmentation complexity, diacritic modeling, and the compounding effect of Devanagari's script characteristics.

Table 12: Qualitative Error Analysis of Nwāchā Munā (underline in prediction column represents the error segment)

No.	Reference	Prediction	Error Analysis
(i)	तस्सकं न्ह्याइपुसेच्वं सीम बत्तिं चैं छन्दत म्याहला च्वंगु tassakaṃ nhyāipusecvaṃ sīma battiṃ caiṃ chandata myāhalā cvaṃgu *It was a lot of fun sitting even in the dim light, singing your song*	तस्सकां न्येयेपुसं सिमापत्ति चैं छंत मां ला च्वं tassakāṃ nyeyepusaṃ simāpatta caiṃ chaṃta māṃ lā cvaṃ	Phonetic substitutions and word distortions; nasal markers and vowels altered within agglutinated structures.
(ii)	दइ धका नं सिल हे daï dhakā naṃ sila he *They say it is known that it will fall*	दइ धका: नं सिल हे daï dhakāḥ naṃ sila he	Insertion of (ः) (Visarga), reflecting minor phonetic misalignment.
(iii)	फोहोर जुलकि लाकां सिले माः phohora julaki lākāṃ sile māḥ *If the shoe is dirty, it must be washed*	पर जुलकि लाका सिने माः para julaki lākā sine māḥ	Nasal suffix deletion and lexical substitution; final (ं) dropped, with minor token distortion.
(iv)	आः थ्व सु āḥ thva su *Now, who is this*	आाख्वसु āākhvasu	Word boundary deletion; independent tokens merged due to agglutinative structure.

5. Conclusion

In this work, we developed resources to help bridge the digital divide for Nepal Bhasha (Newari) by curating a high-quality 5.39-hour transcribed speech corpus, with all transcripts encoded in the Devanagari script. Using this resource, we systematically evaluated cross-lingual transfer learning strategies, benchmarking `NepConformer` along with a multilingual `Whisper-Small` baseline. Our experiments demonstrate substantial gains from supervised fine-tuning and data augmentation, reducing the CER from a high zero-shot baseline to below 18%. The frozen-encoder fine-tuning strategy further enabled stable and effective adaptation in this ultra-low-resource setting. Moreover, shallow fusion with a KenLM n-gram language model enhances lexical regularity and contributes to a reduction in WER, even when character-level improvements remain marginal, indicating improved word-level consistency in decoded outputs. Beyond Nepal Bhasha, our findings indicate that intra-regional transfer learning within South Asian language clusters may offer a computationally efficient pathway for scaling ASR to other endangered and minority languages

Data and Code Availability The training scripts and source code are available on GitHub (https://github.com/ilprl/nwacha-muna). The model weights and audio datasets are hosted on Hugging Face (https://huggingface.co/collections/ilprl-docse/nwacha-muna).

6. Ethics Statement and Limitations

This work offers meaningful societal and technological benefits for the Nepal Bhasha speaking community. By providing an open-source Devanagari speech corpus and establishing strong ASR benchmarks, the resources developed pave the way for voice-driven AI products that can significantly enhance digital accessibility. Furthermore, this research plays an important role in building extensive speech archives, allowing endangered language communities to actively document and preserve their linguistic heritage.

The dataset was collected through voluntary participation from the community, involving native speakers across diverse age groups and genders to ensure representative coverage. For the recorded audio, consent was obtained from all volunteers, and no sensitive or personally identifiable information (PII) was intentionally included in the spoken transcripts. Nonetheless, we acknowledge that models trained on limited data may reflect embedded societal biases or result in acoustic-phonetic misrepresentations, particularly in ultra-low-resource settings. Because of these limitations, the baseline ASR system developed in this study should not be deployed in critical applications without rigorous human verification. The primary objective of this work remains the promotion of linguistic inclusion and digital support for under-resourced language communities.

Despite the promising benefits, this work faces several constraints related to ultra-low-resource settings. Primarily, the curated Nwāchā Munā

dataset, while high-quality, remains relatively small compared to standard ASR benchmarks. Our current evaluation focuses predominantly on read speech and formal sentences, which may not fully capture the acoustic variability, disfluencies, and rapid speaking rates encountered in everyday spontaneous conversations. Also, external n-gram language model utilized during shallow fusion is constrained by domain mismatches and possesses limited coverage of natural, conversational linguistic patterns.

7. Bibliographical References

Shweta Bansal, Shweta Sinha, and Shyam S. Agrawal. 2020. Acoustic-phonetic approach for ASR of less resourced languages using monolingual and cross lingual information. In *Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020)*, pages 167–171. European Language Resources Association (ELRA).

Bharat Bhatta, Basanta Joshi, and Ram Krishna Maharjhan. 2020. Nepali speech recognition using CNN, GRU and CTC. In *Proceedings of the 32nd Conference on Computational Linguistics and Speech Processing (ROCLING 2020)*, pages 238–246. The Association for Computational Linguistics and Chinese Language Processing.

Yao-Fei Cheng, Li-Wei Chen, Hung-Shin Lee, and Hsin-Min Wang. 2024. Exploring the impact of data quantity on asr in extremely low-resource languages.

Thennal D K, Jesin James, Deepa Padmini Gopinath, and Muhammed Ashraf K. 2025. Advocating character error rate for multilingual ASR evaluation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4941–4950, Albuquerque, New Mexico. Association for Computational Linguistics.

Manish Dhakal, Arman Chhetri, Aman Kumar Gupta, Prabin Lamichhane, Suraj Pandey, and Subarna Shakya. 2022. Automatic speech recognition for the nepali language using CNN, bidirectional LSTM and ResNet. In *2022 International Conference on Inventive Computation Technologies (ICICT)*, pages 515–521. IEEE.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2026. *Ethnologue: Languages of the World*, twenty-ninth edition. SIL International, Dallas, Texas. Online version: http://www.ethnologue.com.

Rupak Raj Ghimire, Bal Krishna Bal, and Prakash Poudyal. 2023a. Active learning approach for fine-tuning pre-trained ASR model for a low-resourced language: A case study of Nepali. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 82–89, Goa University, Goa, India. NLP Association of India (NLPAI).

Rupak Raj Ghimire, Bal Krishna Bal, Balaram Prasain, and Prakash Poudyal. 2023b. Pronunciation-aware syllable tokenizer for Nepali automatic speech recognition system. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 36–43, Goa University, Goa, India. NLP Association of India (NLPAI).

Rupak Raj Ghimire, Prakash Poudyal, and Bal Krishna Bal. 2025. Improving accuracy of low-resource asr using rule-based character constituency loss (rbccl). In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*, pages 61–70. Association for Computational Linguistics.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Proceedings of Interspeech 2020*, pages 5036–5040. ISCA.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Tahir Javed, Sumanth Doddapaneni, Abhigyan Raman, Kaushal Santosh Bhogale, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Towards building ASR systems for the next billion users. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*.

Basanta Joshi and Rupesh Shrestha. 2023. Nepali speech recognition using self-attention networks. *International Journal of Innovative Computing, Information and Control*, 19(6):1769–1784.

Marieke Meelen, Alexander O'Neill, and Rolando Coto-Solano. 2024. End-to-end speech recognition for endangered languages of nepal. In *Proceedings of the 7th Workshop on Computational Methods for Endangered Languages (ComputEL-7)*, pages 83–93. Association for Computational Linguistics.

Christopher Moseley, editor. 2010. *Atlas of the World's Languages in Danger*, 3rd edition. UNESCO Publishing, Paris.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Kiran Pantha, Rupak Raj Ghimire, and Bal Krishna Bal. 2025. Speech personalization using parameter efficient fine-tuning for Nepali speakers. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 190–199, Naples, Italy. Unior Press.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Proceedings of Interspeech 2019*, pages 2613–2617. ISCA.

Shishir Paudel, Bal Krishna Bal, and Dhiraj Shrestha. 2023. Large vocabulary continous speech recognition for Nepali language using CNN and transformer. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 328–333, Vienna, Austria. NOVA CLUNL, Portugal.

Post Report. 2024. Nepali, tamang, nepal bhasha official languages of bagmati. *The Kathmandu Post*. Accessed: 2026-02-26.

Jenny Poudel, Ankit Dahal, Rishikesh Kumar Sharma, Rupak Tiwari, Rupak Raj Ghimire, and Bal Krishna Bal. 2025. Nepconformer: A conformer-based nepali automatic speech recognition system. In *International Conference on Computing and Machine Learning*, pages 167–178. Springer.

Lawrence R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

P Regmi, A Dahal, and B Joshi. 2019. Nepali speech recognition using rnn-ctc model. *International Journal of Computer Applications*.

Sunil Regmi and Bal Krishna Bal. 2021. An end-to-end speech recognition for the Nepali language. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 180–185, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Manish K. Ssarma, Avaas Gajurel, Anup Pokhrel, and Basanta Joshi. 2017. Hmm based isolated word nepali speech recognition. In *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 1, pages 1–6. IEEE.

Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu. 2018. A comparison of techniques for language model integration in encoder-decoder speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 369–375. IEEE.

Ashish Vaswani, Llion Jones, Noam Shazeer, Niki Parmar, Aidan N. Gomez, Jakob Uszkoreit, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

From Romanized to Devanagari: Enhancing Nepali Sentiment Analysis with NepaliXlit

Suraj Patel, Kashish Kumari Dhami, Norden Sherpa, Supriya Khadka
Sunway College, Birmingham City University, Kathmandu, Nepal
{suraj_patel_a25, kashish_dhami_a25, norden_23189632, supriya}@sunway.edu.np

Abstract

Romanized Nepali is the dominant medium of social media communication in Nepal, yet most multilingual NLP models are trained on Devanagari, creating a noticeable drop in performance in informal settings. To address this script mismatch, we develop NepaliXlit, a transliteration model fine-tuned from IndicXlit to better handle the phonetic variability of Romanized Nepali. Trained on 2,943 informal word pairs and evaluated on 736 held-out pairs, NepaliXlit improves transliteration accuracy by 8% and reduces character error rate by 11%. We use sentiment analysis as a testbed to understand whether transliteration actually helps downstream NLP tasks. We curate over 6,500 Romanized social media comments and construct a balanced subset of 1,518 manually annotated instances. Baseline experiments show that multilingual encoder models struggle with Romanized input; however, transliterating text into Devanagari using NepaliXlit consistently improves sentiment classification accuracy with mBERT and MuRIL. Comparative evaluation against large language models (LLMs) further reveals that generative models such as Gemini and GPT variants exhibit strong cross-script generalization and outperform encoder-based baselines. Our results indicate that adaptive transliteration enhances conventional multilingual models, while modern LLMs offer a better alternative for multi-script, low-resource settings.

Keywords: Transliteration, Romanized Nepali, Cross-Script NLP, Sentiment Analysis

1. Introduction

The Nepali language, historically referred to as Khas Kura or Gorkhali, is the lingua franca of Nepal and is spoken by millions across South Asia and the global diaspora (Sarkar, 2026). The Devanagari script is the official writing system for Nepali and is deeply embedded in the nation's administrative and literary history (Dahal, 2000). Devanagari exhibits relatively strong grapheme-to-phoneme correspondence and forms the foundation of standard Nepali orthography (Kumar et al., 2010).

However, the digital age has introduced a significant divergence between formal written standards and informal online communication. Despite the availability of native language keyboards, the Roman script has emerged as the dominant mode of digital communication for many Indic languages, including Nepali. This shift is largely driven by the global ubiquity of QWERTY keyboards on mobile devices and computers. As noted by Madhani et al. (2023a), the learning curve associated with native keyboard layouts is often cumbersome for multilingual South Asian users, leading to a strong preference for Romanized input. This results in a vast volume of user-generated content on platforms such as Reddit, Facebook, and Twitter created using Romanized Nepali, a phonetic approximation of the language using Latin characters (Sitoula et al., 2025).

This widespread reliance on Romanized input creates a structural mismatch for Natural Language Processing (NLP). Most state-of-the-art NLP models for Nepali are pre-trained primarily on Devanagari corpora (Pudasaini et al., 2024), meaning that the script gap between user input and model training data presents a substantial barrier to effective text analysis in informal domains. While Devanagari maintains orthographic stability, Romanized Nepali is highly unstructured. Although formal Romanization systems exist, such as those documented by Shrestha (2012), they are rarely adhered to in casual social media discourse. Instead, users employ non-standardized phonetic spellings that vary heavily based on individual preference, creating noisy text that lacks consistent orthographic rules.

For example, in the sentence आज दिन राम्रो छ, meaning "Today is a good day," the word आज ($/\mathrm{a}\text{ː}.\widehat{\mathrm{dz}}\mathrm{ʌ}/$) may appear as *aaja*, *aja*, *aajaa*, or *aaj*. Similarly, छ ($/\widehat{\mathrm{ts}}^{\mathrm{h}}\mathrm{ʌ}/$) may be written as *chha*, *cha*, *chh*, *xa*, *xaa*, or simply *x*. Such variability introduces significant noise and further exacerbates the gap between real-world usage and model training distributions. In addition, the Devanagari script itself presents specific challenges and biases in NLP due to its structural complexity and the linguistic diversity of the South Asian region (Khadka and Bhattarai, 2025). Together, these factors highlight the need for approaches that explicitly address script mismatch in Nepali NLP.

To bridge this gap, transliteration, the conversion of text from one script to another based on

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 115–126
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

phonetic similarity (Madhani et al., 2023a), can be employed as a preprocessing strategy. By converting Romanized inputs into native Devanagari, noisy social media text can be normalized and made compatible with multilingual encoders and large-scale Indic language resources (Baral, 2025). However, existing transliteration tools are not tailored to the informal, phonetically inconsistent Nepali found on social media. As we demonstrate in this work, this limitation necessitates the development of a domain-adapted transliteration model capable of handling non-standard spellings.

To evaluate the efficacy of this approach, we use sentiment analysis as a simple but effective way to test downstream performance. Sentiment analysis is the computational study of people's opinions, sentiments, and emotions toward entities and events (Anderson et al., 2024). The rise of internet usage has generated a large volume of opinionated social media content, making automated sentiment detection valuable for research and industry alike (Pudasaini et al., 2024). Importantly, sentiment analysis also provides a controlled framework for assessing model performance to informal and noisy language (Barbieri et al., 2022). By applying sentiment analysis to both raw Romanized text and transliterated Devanagari text, we systematically evaluate whether bridging the script gap results in improvements in downstream NLP performance in low-resource settings.

Our core contributions are:

- We develop NepaliXlit, a domain-adapted transliteration model fine-tuned from IndicXlit to natively process informal, phonetically variable spellings.
- We compile and annotate a balanced three-class Social Sentiment Analysis dataset of Romanized Nepali social media comments.
- We demonstrate via systematic experiments with mBERT and MuRIL that transliteration consistently improves downstream classification accuracy.
- We evaluate multiple frontier Large Language Models (LLMs) against these datasets, providing a comprehensive cross-script evaluation for Nepali sentiment analysis.

2. Related Work

Prior work relevant to this study falls into four intersecting areas: Nepali transliteration systems, large-scale Indic transliteration models, sentiment analysis for Nepali and multilingual settings, and cross-script modeling challenges. Together, these strands illustrate both the progress made and the gap in handling informal Romanized Nepali.

2.1. Nepali Transliteration

The computational processing of Nepali and Devanagari scripts traditionally focuses on bridging the gap between phonetic representation and standardized digital formats. Early efforts in this domain relied heavily on rule-based systems that applied deterministic character-level mappings (Shrestha, 2012). These systems used handcrafted grapheme-to-phoneme rules to convert between scripts, often employing predefined transliteration tables and finite-state style processing to ensure consistent one-to-one correspondences (Bhalla et al., 2013; De Mel et al., 2025). Similarly, Khanal et al. (2021) utilized a manually curated mapping table to convert printed Nepali text into audible speech. While these rigid mappings work effectively for formal and predictable text, they struggle with the linguistic nuances of colloquial language.

Recent advancements have shifted toward sophisticated normalization pipelines designed to handle the inconsistencies of Romanized Nepali. The NEPTUN framework by Chaudahry et al. (2025) represents a significant step in this direction by integrating phonetic normalization with lexical validation. This system converts Romanized input into Devanagari to verify it against an established Nepali lexicon before producing a standardized Romanized output. By stabilizing the input text, this method significantly enhances the accuracy of downstream tasks such as sentiment classification. Despite these improvements, existing systems remain optimized for either formal text or lexicon-constrained normalization, and do not explicitly address the extreme phonetic variability and non-standard shorthand prevalent in informal Romanized Nepali social media.

2.2. IndicXlit and Its Limitations

The most significant advancement in Indic transliteration came through the Aksharantar project (Yash Madhani and others, 2022), which compiled the largest publicly available transliteration corpus of 26 million word pairs across 21 Indic language pairs. This dataset was used to develop IndicXlit, a multilingual Transformer-based model achieving a 15% accuracy improvement over previous evaluations. IndicXlit has since become the de facto evaluation standard in the field and infrastructure for data generation, producing lower token fertility and better embedding alignment than rule-based systems (Jaavid et al., 2024). However, because Aksharantar is constructed primarily from formal, dictionary-sourced word pairs,

IndicXlit is not optimized for the highly inconsistent spellings characteristic of informal social media input Baruah et al. (2024). Our NepaliXlit model addresses this domain gap by fine-tuning IndicXlit on social media-derived Romanized-Devanagari word pairs, specializing it for the phonetically inconsistent spellings common in casual digital discourse.

2.3. Models Used for Sentiment Analysis

Because our objective is to evaluate whether transliteration improves downstream performance, the choice of sentiment classification models must reflect current standards in Nepali NLP. While early sentiment analysis relied on probabilistic classifiers such as Naïve Bayes and SVMs (Rish et al., 2001; Forman, 2008), and later deep learning architectures such as CNN-BiLSTM hybrids (Hashmi et al., 2024), the field is now dominated by pre-trained Transformer-based models. These models consistently outperform traditional approaches in multilingual and low-resource settings, including Aspect-Based Sentiment Analysis tasks (Narayanaswamy, 2021).

In the Nepali context, Sitoula et al. (2025) demonstrated that multilingual transformers such as XLM-RoBERTa and DistilBERT outperform SVM, Random Forest, CNN, and LSTM baselines on Reddit sentiment classification. This establishes transformer-based multilingual encoders as the appropriate and competitive baseline for Nepali sentiment analysis. Building on this foundation, our work evaluates mBERT (Devlin et al., 2019) and MuRIL (Khanuja et al., 2021) to systematically assess whether transliterating Romanized input into Devanagari improves performance within this model family. We further assess frontier LLMs to examine whether large generative models exhibit stronger cross-script generalization without explicit normalization.

2.4. Cross-Script Modeling

A fundamental structural challenge in Nepali NLP lies in the mismatch between model training data and real-world usage. Most available models are optimized for Devanagari corpora or high-resource multilingual data, whereas informal digital communication is dominated by Romanized Nepali. This asymmetry creates performance degradation when models trained on standardized script distributions encounter phonetically variable Romanized input. Large multilingual encoders are further affected by the "curse of multilinguality" (Xu et al., 2021), where model capacity is diluted across many languages, disproportionately impacting low-resource ones such as Nepali. Within the Nepali ecosystem, NepBERTa (Timilsina et al., 2022) represents a strong monolingual Devanagari model and has shown effectiveness on YouTube comments (Pudasaini et al., 2024). However, it was not designed to process Romanized input. Similarly, NepaliGPT (Pudasaini et al., 2025) was trained on a strictly filtered Devanagari corpus that excluded non-Devanagari tokens, rendering it unsuitable for Romanized or code-mixed settings.

Some work has explored Romanized Nepali directly. For example, Pradhananga and Sah (2023) demonstrated that BERT outperforms traditional classifiers on Romanized reviews. However, these approaches do not address the structural script mismatch between training corpora and informal usage. Rather than training separate models per script, our work investigates whether domain-adapted transliteration can bridge this divide and restore compatibility with existing Devanagari-trained encoders, while also comparing performance against modern LLMs that may inherently generalize across scripts.

3. Methodology

This section outlines our three-phase methodology for sentiment analysis of Romanized Nepali text. First, we detail the architecture and fine-tuning of our transliteration model, NepaliXlit. Next, we describe the dataset creation and annotation process for social media comments. Finally, we present the experimental setup used to evaluate model performance across different script configurations.

3.1. Development of NepaliXlit

The core challenge in processing Nepali social media text is the script gap between informal Romanized input and models trained on Devanagari (Pudasaini et al., 2024). Without normalization, Transformer-based classifiers encounter highly variable Romanized forms, resulting in degraded performance. To address this, we develop *NepaliXlit*, a domain-adapted transliteration model that converts noisy Romanized strings into standardized Devanagari[1]. Following the transliteration-as-normalization paradigm validated by Sampath and Supriya (2024), NepaliXlit is specifically fine-tuned to handle the high phonetic variability of informal "Internet Nepali," mapping non-standard strings to their correct linguistic counterparts. Example mappings are shown in Table 1.

By normalizing these variations at the preprocessing stage, text becomes compatible with downstream Devanagari-trained models such as

[1]NepaliXlit is publicly available on https://github.com/Supriya090/NepaliXlit/tree/master/cli

Romanized	Devanagari	Phonetic (IPA)	English
manxe	मान्छे	/maːn.t͡sʰ e/	person
hunxa	हुन्छ	/hun.t͡sʰ ʌ/	okay / happens
garnaw	गर्न	/gʌr.nʌ/	to do

Table 1: Sample transliteration word mappings from the NepaliXlit training dataset.

Romanized Input	Transliterated Results	English Translation
Priyanka le kati ramro boleko k	प्रियंका ले कति राम्रो बोलेको के	Priyanka speaks so well.
Rekha thapa mam lai leuna paro	रेखा थापा माम लाई ल्युन परो	We need to bring Rekha Thapa Ma'am.
mahina vaisakyo ghar dekhi baahira gako xaina	महिना भइसक्यो घर देखि बाहिर गाको छैन	It's been months since I left the house.
Nepal ma kam garni manche na uha parnu hunxa	नेपाल मा काम गर्नी मान्छे न उहाँ पर्नु हुन्छ	He is the one who works in Nepal.
Priyanka kati choti yo maan jetcheu Mero	प्रियंका कति छोटी यो मान जेट्छेउ मेरो	*Priyanka, how many times will you win my heart?*

Table 2: Qualitative examples of transliteration generated using NepaliXlit. The first four samples demonstrate accurate conversion of informal social media text, while the final sample (italicized) highlights a model error where "jetcheu" was incorrectly mapped to "जेट्छेउ" instead of the contextually appropriate "जित्छौ".

mBERT (Devlin et al., 2019) and MuRIL (Khanuja et al., 2021).

3.1.1. Architecture and Corpus Preparation

We selected IndicXlit (Madhani et al., 2023b) as the base architecture due to its transformer-based encoder-decoder design and native support for Nepali script. To adapt it for informal social media text, we employed a domain-adaptation strategy via fine-tuning, retaining general Indic transliteration capabilities while accommodating the orthographic variability in our dataset (Baruah et al., 2024). The fine-tuning corpus consists of 3,679 parallel Romanized–Devanagari word pairs, sourced from 2,500 social media comments collected across Facebook and Reddit. We prioritized high-frequency terms with observed informal spellings. Each mapping was manually verified by two native Nepali speakers to ensure correctness. Character-level tokenization was applied, with space-separated characters, allowing the model to capture diverse spelling variations without relying on a fixed vocabulary. We partitioned the dataset into 80% training and 20% testing splits. The held-out test set contains informal spellings representative of real-world social media usage, allowing us to directly evaluate domain-specific transliteration performance.

3.1.2. Fine-tuning and Evaluation

We fine-tuned the model using the Fairseq framework (Ott et al., 2019), retaining the sequence-to-sequence Transformer encoder-decoder architecture of IndicXlit. NepaliXlit was trained on the 80% training portion of the Romanized–Devanagari corpus and evaluated on the held-out 20% test split consisting of informal social media word forms. Training was conducted on an NVIDIA GeForce RTX 3060 (8GB).

We evaluated transliteration quality using Top-1 Accuracy and Character Error Rate (CER). Compared to the base IndicXlit model evaluated on the same test set, NepaliXlit achieved a **relative improvement of approximately 8% in Top-1 Accuracy** and a **relative reduction of 11% in CER**. These improvements indicate that domain-specific fine-tuning enhances robustness to noisy and phonetically variable Romanized Nepali text. While this work focuses primarily on the downstream impact of transliteration on sentiment classification, these results confirm that NepaliXlit provides a stronger normalization layer than the base IndicXlit model. Some examples of the transliteration done by NepaliXlit is given in Table 2. A more extensive re-evaluation of transliteration performance with additional metrics and larger evaluation sets is left for future work.

3.2. Sentiment Dataset Creation

3.2.1. Data Collection and Filtering

We collected Romanized Nepali comments from multiple social media platforms, primarily targeting Facebook, followed by YouTube and Instagram. To ensure a high density of Romanized

Nepali text, we specifically focused on content popular among youth audiences. On Facebook, we scraped comments from high-engagement pages such as *Routine of Nepal Banda*[2]. On YouTube, we targeted popular podcasts and vlogs, selecting channels such as *Surakshya KC*[3] and *Priyanka Karki*[4], where viewers frequently interact using Romanized script.

Using the YouTube Comment Exporter extension[5], along with manual extraction for Facebook and Instagram, we conducted 69 collection runs across 69 distinct posts. Although the exporter has a maximum capacity of 100 comments per post, we retrieved an average of 80 comments per run to maintain data quality, resulting in an initial pool of 6,566 raw comments. The raw dataset consisted of a mix of emojis, pure English text, Devanagari Nepali, and multilingual content. We utilized Gemini Pro (Comanici et al., 2025) to filter out non-target content, specifically removing emojis, English-only sentences, and Devanagari text to isolate a cleaned Romanized Nepali dataset.

3.2.2. Sentiment Annotation and Balancing

We manually labeled each comment as Neutral (0), Positive (1), or Negative (2), following standard three-class sentiment analysis practices (Anderson et al., 2024). The annotation was conducted by two students who are active social media users, ensuring they were well-versed in the slang and nuances of modern Romanized Nepali.

Sentiment	Initial Count	Final Count
Neutral	582	515
Positive	510	501
Negative	409	502

Table 3: Sentiment class distribution before and after balancing.

As shown in Table 3, the initial filtering of the 6,500+ comments resulted in a distribution where the negative class was significantly underrepresented (409 samples) compared to neutral and positive classes. To address this imbalance and prevent model bias (Sitoula et al., 2025), we performed a targeted second round of scraping. During this phase, we specifically sought out posts with controversial topics or heated discussions likely to yield negative sentiments. By repeating the collection and cleaning process for these specific sources, we successfully increased the negative sample size to match the other classes. The finalized, balanced dataset consists of 1,518 comments, providing a foundation for evaluating transformer-based and large language models[6].

3.3. Experimental Setup

We designed experiments to compare sentiment analysis performance across Romanized Nepali, Devanagari Nepali, and combined script datasets. We evaluated both fine-tuned Transformer models and LLMs for this comparison.

3.3.1. Transformer Models and Training Configuration

We fine-tuned two Transformer-based encoders, selected for their strengths in multilingual and Indic-specific contexts. Specifically, we used mBERT, pre-trained on 104 languages using masked language modeling (Devlin et al., 2019), and MuRIL, pre-trained on 17 Indic languages along with their transliterated counterparts (Khanuja et al., 2021). MuRIL's architecture is optimized for Devanagari text, making it suitable for transliterated datasets.

All models were trained on datasets consisting of text paired with corresponding sentiment labels. We enforced strict train–test separation to prevent data leakage. Notably, this test set differs from the "held-out" test set used for finetuning IndicXlit. MuRIL was trained exclusively on transliterated Devanagari data, while mBERT was evaluated under three configurations: Romanized only, Devanagari only, and a combination of both scripts (Table 4). All models were trained on Google Colab using GPU acceleration.

Configuration	Train	Test
Romanized only	1200	318
Devanagari only	1200	318
Combined scripts	2400	636

Table 4: Fine-tuning configurations for sentiment classification.

3.3.2. Large Language Models Evaluation

We selected LLMs via OpenRouter[7] for multilingual support and efficiency. We evaluated them

[2] https://www.facebook.com/officialroutineofnepalbanda/

[3] https://www.youtube.com/@SURAKSHYAKCOFFICIAL

[4] https://www.youtube.com/@PriyankaKarki11

[5] https://chromewebstore.google.com/detail/youtube-comment-exporter/hfkfcomfcfeanhckcdohojjjlaagmgpd

[6] Social Sentiment Analysis Dataset is publicly available at https://github.com/anynomousresearch/Data

[7] https://openrouter.ai/

in a zero-shot setting using the following prompt:

> Classify the sentiment of this Nepali comment: {comment}
> Labels: 0 Neutral; 1 Positive; 2 Negative.
> Answer with only the label.

We included Gemini 2.5 Pro (Google DeepMind) (Comanici et al., 2025) for its advanced handling of mixed scripts, effective with both Romanized and Devanagari text. We used LLaMA 3.3 70B Instruct (Meta) (Grattafiori et al., 2024), an open-source instruction-tuned model capable of multilingual tasks, and Qwen3 Max Thinking (Alibaba) (Yang et al., 2025), optimized for reasoning and multilingual processing. Claude Haiku 4.5 (Anthropic) served as a low-cost model suitable for short classification tasks, while Grok 4.1 Fast (xAI) provided speed and efficiency for bulk processing. We further incorporated GPT-OSS 120B (OpenAI) (Agarwal et al., 2025) as an open-source baseline and GPT-5.2 (OpenAI) as a high-performance model for sentiment analysis. This careful selection balanced high-accuracy with cost-efficient bulk classification.

3.3.3. Experimental Workflow

We designed a four-stage pipeline to ensure systematic evaluation and reproducibility. First, we applied Script Standardization, using NepaliXlit to transliterate Romanized comments into Devanagari, which created parallel datasets to minimize orthographic noise. Second, during the Dataset Configuration stage, we organized these outputs into Romanized, Devanagari, and combined subsets with fixed train-test splits. Third, we conducted Transformer Fine-Tuning on mBERT and MuRIL using consistent hyperparameters, ensuring a fair comparison across scripts. Finally, we executed the LLM Evaluation stage, utilizing an asynchronous zero-shot protocol to evaluate frontier models. We employed concurrency controls and batched processing in this final stage, ensuring stable and scalable inference across all dataset variations.

4. Results

This section presents the results we obtained from the experimental configurations described in Section 3. We evaluated performance using overall accuracy as the primary metric, maintaining consistency with prior Nepali sentiment analysis evaluation studies (Sitoula et al., 2025; Pudasaini et al., 2024). We report the results separately for encoder-based models and LLMs.

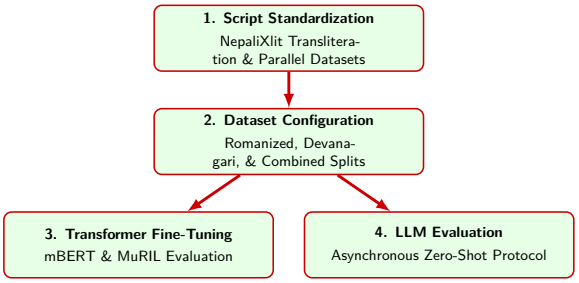


Figure 1: Four-stage experimental workflow. The pipeline sequentially advances from script standardization and dataset configuration to transformer fine-tuning and LLM evaluation phases.

Model	Input Script	Baseline (%)	Finetuned (%)
mBERT	Romanized	35.85	**49.69**
mBERT	Devanagari	34.91	**53.77**
mBERT	Combined	36.16	**50.00**
MuRIL	Devanagari	31.76	**60.06**

Table 5: Comparison of baseline and fine-tuned model accuracy for the sentiment analysis task across different input scripts. Bold values indicate the performance improvements after fine-tuning.

4.1. Impact of Script Normalisation on Encoder-Based Model Performance

Table 5 presents a comprehensive comparison of encoder-based model performance across different input scripts and normalisation strategies. In the baseline configuration lacking script normalisation, both mBERT and MuRIL exhibited notably low accuracy on Romanized input, scoring 35.85% and 31.76% respectively. These results highlight the significant difficulty these models face in processing informal Romanized Nepali text.

We evaluated NepaliXlit on a held-out transliteration test set, where it increased Top-1 accuracy by 8% and reduced character error rate by approximately 11% compared to the IndicXlit baseline. Applying this transliteration as a preprocessing step created standardized Devanagari input, which improved encoder performance across all configurations. Notably, the Devanagari-only transliterated configuration for mBERT achieved higher accuracy than the Romanized-only setup, confirming the benefits of script standardisation.

MuRIL achieved the strongest overall performance among encoder models, reaching 60.06% accuracy on the transliterated dataset. Despite these overall improvements, classifying the neutral sentiment class remained the most significant challenge across all configurations, as detailed in Table 6.

Model	Training Configuration	Overall (%)	Neutral (%)	Positive (%)	Negative (%)
mBERT	Romanized-only	49.69	40.87	51.49	57.84
mBERT	Devanagari-only (Translit.)	53.77	50.43	51.49	59.80
mBERT	Combined (Roman. + Deva.)	50.00	33.04	65.35	53.92
MuRIL	Devanagari (Translit.)	60.06	38.26	74.26	70.59

Table 6: Accuracy performance comparison after transliteration-based fine-tuning. We can see that neutral is consistently the worst performing label (denoted by an underline).

Model	Romanized				Devanagari			
	Overall (%)	Neu. (%)	Pos. (%)	Neg. (%)	Overall (%)	Neu. (%)	Pos. (%)	Neg. (%)
Gemini-2.5-Pro	**73.14**	37.39	**97.03**	91.40	68.27	31.30	**92.93**	86.73
GPT-5.2	72.06	30.36	96.04	**94.12**	**69.52**	34.78	90.10	**88.89**
GPT-OSS-120B	63.92	**38.94**	84.16	71.57	62.10	**40.87**	75.26	73.53
Grok-4.1-Fast	61.95	20.00	83.17	88.24	60.26	19.13	80.00	88.24
Qwen3-Max	60.58	22.61	85.26	80.39	60.90	28.44	81.19	75.49

Table 7: Sentiment analysis performance across scripts (bold indicates highest value per column).

4.2. LLM Performance Results

Table 7 presents the zero-shot evaluation results we obtained by evaluating multiple large language models on Romanized and transliterated Devanagari datasets. Gemini-2.5-Pro achieved the highest overall accuracy of 73.14% on Romanized input, and GPT-5.2 followed closely at 72.06%. Furthermore, GPT-5.2 demonstrated strong cross-script consistency by maintaining a competitive performance of 69.52% on Devanagari input. This strong cross-script performance of frontier LLMs supports the broader findings of Azam et al. (2025), who concluded that large general-purpose language models exhibit stronger cross-script generalization than specialized fine-tuned models for Indic languages. The persistent difficulty with neutral sentiment classification across all LLMs mirrors the pattern we observed in encoder-based models and aligns directly with the research of Sitoula et al. (2025).

4.3. Comparative Analysis

Our results reveal that LLMs substantially outperform encoder-based models in overall accuracy for Nepali sentiment analysis. Furthermore, a clear divergence in optimal preprocessing strategies emerged between the two model families. While transliteration to Devanagari proved essential for maximizing the performance of encoder-based models like mBERT and MuRIL, frontier LLMs exhibited superior performance when processing the original Romanized input.

As illustrated in Figure 2, the highest-performing models Gemini-2.5-Pro and GPT-5.2 achieved their peak accuracy on Romanized text, experiencing a performance degradation when evaluated on transliterated Devanagari. These findings indicate that while adaptive transliteration remains a critical bridge for conventional multilingual encoders (Madhani et al., 2023b), it may be less critical for modern generative models that demonstrate stronger cross-script generalization (Azam et al., 2025). This divergence suggests that the "script gap" affects discriminative and generative architectures differently, a trend we explore further in the Discussion.



Figure 2: Zero-shot performance of LLMs across Romanized and Devanagari datasets. Gemini-2.5-Pro and GPT-5.2 outperform other models; most LLMs show higher accuracy on Romanized input.

5. Discussion

Our experimental results offer meaningful insights into handling informal, Romanized Nepali within modern NLP pipelines. Our findings validate the hypothesis that transliteration remains a vital preprocessing step for encoder-based models. Simultaneously, we reveal that state-of-the-art LLMs

possess an inherent adaptability to script variation that challenges traditional adaptation strategies. This section discusses the implications of these findings within the context of existing literature and identifies key areas for future research.

5.1. The Efficacy of Domain-Adapted Transliteration

The significant performance gain observed in encoder-based models following transliteration underscores the persistence of the script gap in low-resource languages. The difficulty standard multilingual encoders face with raw Romanized input suggests a lack of sufficient exposure to informal Romanized orthographies during pre-training, likely due to the historical dominance of Devanagari script in their training corpora (Khanuja et al., 2021). Furthermore, non-standard spellings typical of social media platforms cause severe token fragmentation. In these instances, the model fails to recognize informal Romanized words as Nepali, treating them instead as out-of-vocabulary or noise (Rust et al., 2021). By utilizing NepaliXlit to convert text into standardized Devanagari, we allow the model to operate within its intended linguistic and structural environment, leading to the observed jump in accuracy. These results extend the arguments of Madhani et al. (2023b), who posited that effective transliteration tools are a prerequisite for Indic NLP. However, our findings specifically demonstrate that generalist models like IndicXlit are often insufficient for the "noisy" social media domain. Consistent with Baruah et al. (2024), our success confirms that effective domain adaptation requires more than simple script conversion; it requires the normalization of sociolect-driven spellings that characterize modern digital communication in Nepal (Sitoula et al., 2025).

5.2. The Shift from Fine-Tuning to Zero-Shot LLMs

The fact that Gemini-2.5-Pro outperformed the best fine-tuned encoder (MuRIL) by over 13 percentage points without task-specific training signals a paradigm shift for low-resource language tasks. This outcome supports the broader observation by Azam et al. (2025) that large general-purpose language models exhibit stronger cross-script generalization than specialized fine-tuned models.

Building upon our analysis in Section 4.3, we attribute the LLMs' native proficiency with Romanized text to three primary factors. First, internet-scale LLMs are extensively pre-trained on informal social media data (Shi et al., 2023), making them inherently robust to Romanized Indic text compared to encoders trained on formal Wikipedia corpora. Second, modern LLM tokenizers are heavily biased toward Latin scripts; in contrast, Devanagari script often suffers from token fragmentation, which can degrade semantic processing (Kanjirangat et al., 2025). Finally, evaluating LLMs directly on Romanized text avoids the error propagation inherent in any intermediate transliteration pipeline.

However, it is important to qualify these findings to avoid overgeneralization. While transliteration appears redundant for high-capacity frontier models in this informal domain, it remains a vital bridge for deploying specialized encoder models (Sampath and Supriya, 2024). Smaller models like MuRIL and mBERT remain necessary for tasks requiring high speed and lower costs, especially in localized technology projects in Nepal where hardware resources are limited (Schwartz et al., 2020). Therefore, while frontier LLMs offer a streamlined, preprocessing-free alternative, domain-adapted tools like NepaliXlit remain essential for maintaining a sustainable and diverse NLP ecosystem.

5.3. The Ambiguity of Neutral Sentiment in Low-Resource Contexts

Across all model architectures and script configurations, the neutral sentiment class consistently presented the greatest classification challenge. This systematic difficulty mirrors the findings of Sitoula et al. (2025), who observed similar ambiguity in Nepali Reddit data. We attribute this to the linguistic nature of neutral social media comments, which often lack explicit sentiment-bearing lexicons and rely heavily on context-dependent interpretation. Current models fail to capture the pragmatic cues that differentiate informational statements from mildly positive or negative expressions in Nepali. We suggest that future annotation efforts prioritize inter-annotator agreement metrics, such as Cohen's Kappa, to better quantify the inherent subjectivity of the neutral class (Anderson et al., 2024). Furthermore, researchers might yield more reliable outcomes for neutral instances by adopting a more granular annotation scheme or utilizing emotion detection rather than broad sentiment polarity (Pudasaini et al., 2024).

6. Limitations and Future Directions

While this study establishes a new baseline for Romanized Nepali sentiment analysis, several limitations remain. First, the dataset size of 1,518 annotated comments, though balanced, is relatively small for training deep transformer models from scratch, which typically require large-scale data to generalize effectively and avoid overfitting (Devlin et al., 2019). Second, our evaluation

of LLMs was limited to zero-shot prompting; incorporating few-shot or chain-of-thought prompting could better leverage the reasoning capabilities of advanced models such as GPT-5.2 and Qwen3-Max (Azam et al., 2025). Third, the study focuses exclusively on monolingual Romanized and Devanagari Nepali text, explicitly excluding code-mixed Nepali-English data during preprocessing. While this ensured a controlled experimental setup for analyzing transliteration, it limits ecological validity, as code-mixing is highly prevalent in real-world Nepali digital communication. Additionally, we did not empirically measure token fragmentation across scripts, instead relying on assumptions from prior studies that Devanagari text may exhibit higher fragmentation; a systematic analysis could reveal important insights into model performance differences.

Future work should address these limitations by scaling the dataset, incorporating more advanced prompting strategies, and extending the framework to handle code-mixed Nepali-English text. Furthermore, evaluating tokenization behavior across scripts and expanding the pipeline to other downstream tasks such as hate speech detection (Khadka et al., 2025), named entity recognition and text-to-speech (Khadka et al., 2023) would help validate the broader applicability of the NepaliXlit tool in real-world scenarios.

7. Conclusion

This paper addresses the challenge of sentiment analysis on informal Romanized Nepali social media text. We introduce a balanced dataset of 1,518 annotated comments and develop NepaliXlit, a domain-adapted transliteration model fine-tuned for sociolect-driven spellings. Our evaluation demonstrates that bridging the script gap through transliteration consistently improves encoder-based sentiment classification. Specifically, MuRIL achieved 60.06% accuracy on transliterated Devanagari data. However, comprehensive zero-shot evaluation reveals that frontier LLMs substantially outperform fine-tuned encoders directly on Romanized input. Notably, Gemini-2.5-Pro reached 73.14% accuracy, suggesting inherently adaptable cross-script representations without task-specific adaptation. Despite these advancements, accurately classifying neutral sentiment remains a persistent challenge across all model architectures. Ultimately, specialized tools like NepaliXlit fill a critical gap for low-resource encoder pipelines. Yet, the strong performance of frontier LLMs indicates a shift in how systems will process informal digital communication in the future. The dataset, model, and evaluation results we present in this work serve as a foundational resource for advancing multilingual NLP and cross-script sentiment analysis in underserved contexts.

8. Ethical Considerations

This study was conducted with strict adherence to ethical standards regarding data privacy and research integrity. The dataset consists of publicly available social media comments, and to ensure the anonymity of users, all personally identifiable information, such as usernames, profile links, and specific mentions, was removed during the preprocessing phase. Furthermore, the manual annotation process was conducted objectively to minimize demographic or linguistic bias. The fine-tuning of the NepaliXlit model and the analysis of LLMs were performed for academic research purposes, ensuring that the results are used to improve linguistic accessibility for low-resource languages without promoting harmful or biased sentiment classification.

9. Data/Code Availability Statement

All the data and code supporting the findings of this study is publicly available on GitHub.

10. References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Tess Anderson, Sayani Sarkar, and Robert Kelley. 2024. Analyzing public sentiment on sustainability: A comprehensive review and application of sentiment analysis techniques. *Natural Language Processing Journal*, 8:100097.

Gulfarogh Azam, Mohd Sadique, Saif Ali, Mohammad Nadeem, Erik Cambria, Shahab Saquib Sohail, and Mohammad Sultan Alam. 2025. Beyond specialization: Benchmarking llms for transliteration of indian languages. *arXiv preprint arXiv:2505.19851*.

Dipesh Baral. 2025. Comparing labse with contrastively and soft-label fine-tuned mbert models for semantic search over a nepali knowledge base. *International Journal on Engineering Technology*, 3(1):146–155.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. Xlm-t: Multilingual

language models in twitter for sentiment analysis and beyond. In *Proceedings of the thirteenth language resources and evaluation conference*, pages 258–266.

Hemanta Baruah, Sanasam Ranbir Singh, and Priyankoo Sarmah. 2024. Assamesebacktranslit: Back transliteration of romanized assamese social media text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1627–1637.

Deepti Bhalla, Nisheeth Joshi, and Iti Mathur. 2013. Rule based transliteration scheme for english to punjabi. *arXiv preprint arXiv:1307.4300*.

Chandra Prakash Chaudahry, Basanta Joshi, Aman Shakya, and Santosh Giri. 2025. Neptun: Normalization for romanized nepali sentiment analysis. *Journal of Himalaya College of Engineering*, 2(1).

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Ram Kumar Dahal. 2000. Language polics innepal. *Contributions to Nepalese Studies*, 27(2):155–190.

Yomal De Mel, Kasun Wickramasinghe, Nisansa De Silva, and Surangika Ranathunga. 2025. Sinhala transliteration: a comparative analysis between rule-based and seq2seq approaches. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 166–173.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

George Forman. 2008. Bns feature scaling: an improved representation over tf-idf for svm text classification. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 263–270.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ehtesham Hashmi, Sule Yildirim Yayilgan, Ibrahim A Hameed, Muhammad Mudassar Yamin, Mohib Ullah, and Mohamed Abomhara. 2024. Enhancing multilingual hate speech detection: From language-specific insights to cross-linguistic integration. *IEEE Access*, 12:121507–121537.

J Jaavid, Raj Dabre, M Aswanth, Jay Gala, Thanmay Jayakumar, Ratish Puduppully, and Anoop Kunchukuttan. 2024. Romansetu: Efficiently unlocking multilingual capabilities of large language models via romanization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15593–15615.

Vani Kanjirangat, Tanja Samardzic, Ljiljana Dolamic, and Fabio Rinaldi. 2025. Tokenization and representation biases in multilingual models on dialectal nlp tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24003–24021.

Pilot Khadka, Ankit Bk, Ashish Acharya, Bikram Kc, Sandesh Shrestha, and Rabin Thapa. 2025. Nepali transformers@ nlu of devanagari script languages 2025: Detection of language, hate speech and targets. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 314–319.

Supriya Khadka and Bijayan Bhattarai. 2025. Gender bias in nepali-english machine translation: A comparison of llms and existing mt systems. In *Proceedings of the 6th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 75–82.

Supriya Khadka, Ranju G.C., Prabin Paudel, Rahul Shah, and Basanta Joshi. 2023. Nepali text-to-speech synthesis using tacotron2 for mel-spectrogram generation. In *SIGUL 2023, 2nd Annual Meeting of the Special Interest Group on Under-resourced Languages: a Satellite Workshop of Interspeech 2023*.

Samir Khanal, Ranjan Paudel, and Rajendra Chataut. 2021. Nepali printed text to speech with accurate transliterated form. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 12(5):97–101.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Uttam Kumar, Tanusree Das, Raju S Bapi, Prakash Padakannaya, R Malatesha Joshi, and Nandini C Singh. 2010. Reading different orthographies: an fmri study of phrase reading in hindi–english bilinguals. *Reading and Writing*, 23(2):239–255.

Yash Madhani, Mitesh M Khapra, and Anoop Kunchukuttan. 2023a. Bhasa-abhijnaanam: Native-script and romanized language identification for 22 indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 816–826.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. 2023b. Aksharantar: Open indic-language transliteration datasets and models for the next billion users. In *Findings of the association for computational linguistics: Emnlp 2023*, pages 40–57.

Gagan Reddy Narayanaswamy. 2021. *Exploiting BERT and RoBERTa to Improve Performance for Aspect Based Sentiment Analysis*. Dissertation, Technological University Dublin, Dublin, Ireland.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (Demonstrations)*, pages 48–53.

Abhash Pradhananga and Anand Kumar Sah. 2023. Transformer-based deep learning models for sentiment analysis in romanized nepali: a comparative investigation of bert and roberta. In *Proceedings of 14th IOE graduate conference*, volume 14.

Shushanta Pudasaini, Sunil Ghimire, Prabhat Ale, Aman Shakya, Prakriti Paudel, and Basanta Joshi. 2024. Application of nepali large language models to improve sentiment analysis. In *Proceedings of the 2024 7th International Conference on Computers in Management and Business*, pages 144–150.

Shushanta Pudasaini, Aman Shakya, Siddhartha Shrestha, Sahil Bhatta, Sunil Thapa, and Sushmita Palikhe. 2025. Nepaligpt: A generative language model for the nepali language. *arXiv preprint arXiv:2506.16399*.

Irina Rish et al. 2001. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. Seattle, USA.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135.

Koyyalagunta Krishna Sampath and M Supriya. 2024. Transformer based sentiment analysis on code mixed data. *Procedia Computer Science*, 233:682–691.

Pintu Sarkar. 2026. The evolution of the nepali language: A brief history. *International Journal of Creative Research Thoughts (IJCRT)*, 14(1):a874–a880. Available at: http://www.ijcrt.org/papers/IJCRT2601109.pdf.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Communications of the ACM*, 63(12):54–63.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Suresh Man Shrestha. 2012. Transliteration system for nepali language. *Journal on Geoinformatics, Nepal*, 11:37–41.

Sameer Sitoula, Tej Bahadur Shahi, Laxmi Prasad Bhatt, Anisha Pokhrel, and Arjun Neupane. 2025. Nepemo: A multi-label emotion and sentiment analysis on nepali reddit with linguistic insights and temporal trends. *arXiv preprint arXiv:2512.22823*.

Sulav Timilsina, Milan Gautam, and Binod Bhattarai. 2022. Nepberta: Nepali language model trained in a large corpus. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 273–284.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine

translation. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6663–6675.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

11. Language Resource References

Yash Madhani and others. 2022. *Aksharantar: Open Transliteration Tools for the Next Billion Users*. arXiv. Language Resource.

Why Does Low-Rank Adaptation Work for Hindi-English Code-Mixing? A Geometric Analysis

Shashank Vishwakarma, Rakesh Kumar
Gautam Buddha University,
Greater Noida, India
shakuvish07@gmail.com, rakesh.kumar@gbu.ac.in

Abstract

Low-Rank Adaptation (LoRA) enables efficient fine-tuning of large language models, yet why it works particularly well for code-mixed text remains unexplained. We propose that LoRA's efficiency stems from geometric structure in multilingual pre-trained models: code-mixed embeddings concentrate in low-dimensional cross-lingual subspaces. Through spectral analysis of mBERT and MuRIL on Hindi-English (Hinglish) data, we establish that pre-trained attention weights have effective ranks of 437–441, while LoRA updates ($r = 4, 8, 16$) exhibit ranks of 2.1–5.9—a 136$\times$ average compression. Cross-lingual geometry measured via Centered Kernel Alignment (CKA) shows Hinglish embeddings align strongly with Hindi (CKA=0.279) but weakly with English (0.093), compared to a monolingual baseline of 0.074. Statistical tests (Wilcoxon $p < 10^{-19}$) and permutation ablations confirm these differences are robust. We interpret the convergence of geometric overlap (3.77$\times$ baseline) and empirical compression (136$\times$) as evidence that low-rank adaptation exploits pre-existing multilingual structure. Findings are demonstrated on token-level language identification; extensions to other language pairs and tasks remain open questions.

Keywords: code-mixing, low-rank adaptation, multilingual models, geometric analysis, representation geometry

1. Introduction

Code-mixing is the phenomenon where speakers alternate between two or more languages within a single utterance. It is pervasive in multilingual communities worldwide. Hindi-English code-mixing (Hinglish) alone occurs in hundreds of millions of social media posts, conversational transcripts, and user-generated content annually. Despite this prevalence, NLP systems trained primarily on monolingual data perform poorly on code-mixed text, necessitating domain-specific adaptation.

Parameter-efficient fine-tuning (PEFT) methods, particularly Low-Rank Adaptation (Hu et al., 2022), have emerged as effective techniques for adapting large pre-trained models to code-mixed tasks. LoRA constrains weight updates to low-rank matrices, typically with ranks $r = 8$–16, achieving performance comparable to full fine-tuning while training only 0.1–1% of model parameters. Empirical studies report that LoRA performs particularly well on multilingual and code-mixed benchmarks, often with lower ranks than required for monolingual tasks (Pfeiffer et al., 2021). However, the mechanisms underlying this efficiency remain unclear. Prior work has characterized LoRA's optimization dynamics and parameter budgets (Hu et al., 2022), but has not explained why code-mixed text enables such substantial rank reduction.

We currently lack a principled understanding of when and why low-rank adaptation suffices. Does LoRA's effectiveness for code-mixing reflect properties of the task, the data, or the representational geometry of the pre-trained model? Can optimal rank be predicted from measurable properties of the adaptation domain? Without mechanistic insight, rank selection remains an empirical trial-and-error process, and generalization to new language pairs or tasks is uncertain.

Hypothesis. We propose that LoRA's efficiency for code-mixing arises from geometric structure in multilingual pre-trained models. Specifically, we hypothesize that code-mixed text induces cross-lingual subspace overlap, such that code-mixed representations occupy a low-dimensional manifold within the pre-trained representation space. If this manifold is well represented during pre-training, then fine-tuning updates need not learn new high-dimensional representations but can instead refine existing structure using low-rank perturbations. This contrasts with monolingual adaptation, where task-specific updates may require access to larger representational capacity.

To test this hypothesis, we adopt a geometric perspective combining spectral analysis and representation similarity metrics. We analyze pre-trained attention weights to establish baseline dimensionality, measure cross-lingual subspace alignment using CKA (Kornblith et al., 2019), and compute effective ranks of learned LoRA updates to quantify empirical compression. By connecting geometric structure (subspace overlap) to adaptation efficiency (low-rank sufficiency), we provide a mechanistic account of LoRA's effectiveness for code-mixing.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 127–136
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

Scope and Contributions. We focus on Hindi-English code-mixing (Hinglish) in the COMI-LINGUA corpus Sheth et al. (2025), with Hindi as the matrix language. Our task is token-level language identification using BERT-based encoder models, specifically mBERT (Devlin et al., 2019) and MuRIL (Khanuja et al., 2021). We do not claim generalization to all language pairs, code-mixing configurations, or NLP tasks; rather, we provide an analysis framework applicable to other scenarios. Our contributions are:

- Empirical characterization of intrinsic dimensionality in pre-trained multilingual models, with effective ranks of 437–441 across attention weights and approximately 90% of spectral energy concentrated in $\sim$301 dimensions.
- Geometric evidence for cross-lingual subspace overlap: Hinglish embeddings exhibit 3.77$\times$ stronger alignment with Hindi (CKA=0.279) than a Hindi-English monolingual baseline (CKA=0.074), confirmed via permutation tests.
- Spectral analysis of trained LoRA adapters, showing effective ranks of 2.1–5.9 (mean=3.9), corresponding to a 136$\times$ average compression factor relative to pre-trained weights, with statistical validation (Wilcoxon test, $p < 10^{-19}$).
- A methodological framework connecting representation geometry to adaptation efficiency, demonstrating convergent evidence across spectral decomposition, cross-lingual similarity metrics, and empirical rank measurements.

Our analysis is strictly descriptive for the Hinglish language identification setting but yields testable predictions: language pairs with higher pre-trained cross-lingual alignment should permit lower LoRA ranks. We discuss limitations, including focus on a single language pair, a single task, and modest dataset size (1,666 samples). Extensions to other language pairs, sentence-level tasks, and generation scenarios remain future work.

2. Related Work

2.1. Parameter-Efficient Fine-Tuning and LoRA

Low-Rank Adaptation (Hu et al., 2022) constrains fine-tuning updates to low-rank matrices $\Delta W = BA$, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$ with $r \ll d$. This reduces trainable parameters by up to four orders of magnitude while maintaining task performance comparable to full fine-tuning. Empirical evaluations demonstrate LoRA's effectiveness across NLP tasks (Dettmers et al., 2023), vision-language models (Liu et al., 2023), and multilingual settings (Dang et al., 2024), but focus primarily on task performance rather than mechanistic explanations for why low-rank updates suffice.

Concurrent PEFT methods include adapter layers (Houlsby et al., 2019), prefix-tuning (Li and Liang, 2021), and BitFit (Ben Zaken et al., 2022). Comparative studies (He et al., 2022) show LoRA achieves favorable efficiency-performance trade-offs for large models. Recent work has analyzed LoRA's optimization dynamics (Malladi et al., 2023), showing low-rank constraints induce implicit regularization. These analyses explain how LoRA optimizes but not why certain domains, such as code-mixing, permit substantially lower ranks. Our geometric analysis connects domain properties to rank requirements.

2.2. Code-Mixing and Multilingual NLP

Code-mixing research has established linguistic theories of matrix-embedded language structure (Myers-Scotton, 1993; Muysken, 2000) and computational models for code-switching detection (Solorio et al., 2014; Aguilar et al., 2020). For Hindi-English, prior work includes language identification (Rijhwani et al., 2017; Khanuja et al., 2020), sentiment analysis (Prabhu et al., 2016), and named entity recognition (Singh et al., 2018). Shared tasks such as GLUECoS (Khanuja et al., 2020) and datasets such as LinCE (Aguilar et al., 2020) have standardized evaluation.

Several studies observe that multilingual models transfer effectively to code-mixed data with minimal fine-tuning (Winata et al., 2021; Doğruöz et al., 2021), often attributing this to generic "multilingual knowledge" without quantifying mechanisms. Gupta et al. (2020) show that synthetic code-mixing during pre-training improves downstream performance, suggesting pre-trained representations already encode cross-lingual structure. However, these works do not analyze representational geometry or connect transfer behavior to low-rank adaptation efficiency.

2.3. Representation Geometry and Spectral Analysis

Canonical Correlation Analysis (CCA) (Raghu et al., 2017) and CKA (Kornblith et al., 2019) have been widely used to compare representations across layers, models, and modalities. Morcos et al. (2018) show CCA reveals shared structure in independently trained networks, while Nguyen et al. (2021) use CKA to study cross-lingual alignment in multilingual models. These studies focus on descriptive characterization rather than linking geometry to adaptation efficiency.

Spectral analysis of neural network weights has revealed low-rank structure in trained models (Martin and Mahoney, 2019; Sharma et al., 2020). Aghajanyan et al. (2021) demonstrate that fine-tuning predominantly occurs in low-dimensional

subspaces, introducing intrinsic dimensionality as the minimal subspace required for task performance. Their approach estimates intrinsic dimension via random projections but does not relate it to pre-trained geometric structure or cross-lingual properties. Recent work by Biderman et al. (2024) analyzes LoRA using singular value decomposition, showing learned updates align with low-frequency components of pre-trained weights. These studies do not address why code-mixing specifically enables stronger alignment or lower rank requirements than monolingual tasks.

Our Contribution. Existing literature establishes that (1) LoRA is effective for code-mixing, (2) multilingual models transfer well to code-mixed data, and (3) neural representations exhibit structured geometry. However, these strands remain disconnected: prior work does not link cross-lingual geometric properties, such as subspace overlap measured via CKA, to LoRA rank requirements for code-mixing. We demonstrate that pre-trained cross-lingual alignment predicts and explains low-rank sufficiency through convergent geometric and spectral evidence. Our findings complement optimization analyses (Malladi et al., 2023) that explain how LoRA converges by explaining why convergence requires low rank for code-mixing specifically.

3. Methodology

3.1. Dataset and Task

We use the COMI-LINGUA corpus Sheth et al. (2025), a Hindi-English code-mixed dataset with token-level language identification annotations. The corpus consists of naturally occurring social media text, where Hindi (Devanagari script) functions as the matrix language with embedded English (Latin script) spans. We use the language identification (LID) configuration, which provides three-way token-level labels: {hi, en, other}.

To ensure data quality and controlled language mixing, we apply inclusion criteria: (1) sentences contain $\geq$5 tokens, (2) $\geq$2 Hindi and $\geq$2 English tokens present, (3) Hindi token proportion between 30–70% to exclude near-monolingual samples, and (4) maximum sequence length 128 tokens. We exclude sentences where all English tokens have length $\leq$3 characters (typically abbreviations). Sentences with more than 95% tokens from a single language are segregated into monolingual control sets (Hindi-only and English-only) for cross-lingual geometry analysis. After filtering, the dataset contains 1,666 samples, split deterministically (seed=42) into training (80%), development (10%), and test (10%) partitions (Table 1).

Table 1: Dataset statistics for COMI-LINGUA after filtering.

Split	Samples	Avg. Length (tokens)	Avg. Hindi (%)
Train	1332	12.8	61.2
Dev	166	13.1	61.7
Test	168	12.6	61.2
Total	1666	12.8	61.2

Language identification is formulated as token-level sequence labeling. Each input token receives one of three labels: Hindi (hi), English (en), or other. We report macro-averaged F1 as the primary metric to ensure balanced treatment of all classes, given that Hindi tokens constitute approximately 61% of the corpus.

3.2. Model Selection

We analyze two multilingual transformer encoder models: `bert-base-multilingual-cased` (Devlin et al., 2019) and `google/muril-base-cased` (Khanuja et al., 2021). Both models have 12 transformer layers with hidden dimensionality 768. mBERT serves as a general multilingual baseline, while MuRIL represents an Indic-language-specialized model, allowing assessment of whether observed phenomena generalize across different pre-training objectives and data distributions.

3.3. Spectral Analysis of Pre-trained Weights

We extract Query and Key projection matrices ($W \in \mathbb{R}^{768\times768}$) from all 12 transformer layers, as these are the primary targets of LoRA adaptation in encoder-only architectures. Value matrices are included during LoRA fine-tuning (Section 3.5) but excluded from baseline spectral analysis. For each matrix W, we compute the full singular value decomposition $W = U\Sigma V^\top$ and retain all 768 singular values.

We quantify intrinsic dimensionality using the entropy-based effective rank:

$$r_{\text{eff}} = \frac{\left(\sum_i \sigma_i\right)^2}{\sum_i \sigma_i^2}, \tag{1}$$

where σ_i denotes singular values in descending order. A rank-k matrix yields $r_{\text{eff}} = k$, while a full-rank matrix with uniform singular values yields $r_{\text{eff}} \approx d$. As a complementary measure, we compute the 90% energy rank:

$$r_{90} = \min\left\{k : \frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{i=1}^{d}\sigma_i^2} \geq 0.9\right\}. \tag{2}$$

3.4. Cross-Lingual Representation Geometry

For geometric analysis, we extract sentence-level representations from the final hidden layer using mean pooling over token embeddings, excluding padding and weighted by attention masks. We sample 300 instances for each condition: (1) Hinglish (code-mixed), (2) Hindi-only tokens extracted from code-mixed sentences, and (3) English-only tokens extracted from the same sentences. This ensures controlled comparison, differing only in retained tokens.

We use CKA (Kornblith et al., 2019) to measure cross-lingual subspace similarity. Given representations $X, Y \in \mathbb{R}^{n \times d}$, CKA is:

$$\text{CKA}(X, Y) = \frac{\text{HSIC}(X, Y)}{\sqrt{\text{HSIC}(X, X)\,\text{HSIC}(Y, Y)}}, \quad (3)$$

where HSIC is the Hilbert-Schmidt Independence Criterion computed using linear kernels:

$$\text{HSIC}(X, Y) = \frac{1}{n^2}\text{tr}(KHLH), \quad (4)$$

with $K = XX^\top$, $L = YY^\top$, and $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. CKA is invariant to orthogonal transformations and isotropic scaling. We compute 95% bootstrap confidence intervals using 1,000 resampling iterations (seed=42).

To verify that observed CKA values reflect structured cross-lingual overlap rather than chance alignment, we conduct a permutation test by randomly shuffling row correspondence between Hinglish and Hindi embeddings while preserving individual representations. CCA is excluded due to saturation artifacts (>0.99 for all pairs), Procrustes distance due to scale sensitivity, and PCA visualizations explain <25% of variance.

3.5. LoRA Fine-tuning

We train Low-Rank Adaptation (Hu et al., 2022) adapters end-to-end on the token-level LID task using mBERT as the base model. We evaluate ranks $r \in \{4, 8, 16\}$ with scaling factor $\alpha = 2r$ (ratio $\alpha/r = 2$). Target modules include Query, Key, and Value projection layers. LoRA dropout is 0.1, and no bias terms are trained.

We use AdamW optimizer (Loshchilov and Hutter, 2017) with learning rate 2×10^{-4}, weight decay 0.01, and batch size 16. Training proceeds for three epochs with early stopping based on development-set macro F1. All experiments use seed 42, and mixed-precision training (fp16) is enabled when supported.

After training, we extract learned LoRA matrices $A \in \mathbb{R}^{r \times 768}$ and $B \in \mathbb{R}^{768 \times r}$. The effective update matrix is:

$$\Delta W = (\alpha/r)BA = 2BA, \quad (5)$$

Table 2: Effective rank and 90% energy rank for pre-trained Query and Key matrices. (Std: Standard Deviation across matrices)

Model	Mean Eff. Rank	Std	Mean 90% Rank
mBERT	436.8	33.5	300.5
MuRIL	441.0	31.4	302.5
Overall	438.9	32.5	301.5

incorporating inference-time scaling. Singular value decomposition is applied to each ΔW, and effective rank and 90% energy rank are computed using Equations 1 and 2. This yields 36 matrices per rank (12 layers $\times$ 3 modules), totaling 108 matrices.

We use the Wilcoxon signed-rank test to compare effective ranks of pre-trained weights ($n = 48$ matrices from mBERT and MuRIL Query/Key) with LoRA effective ranks ($n = 108$ matrices across all ranks and modules). This non-parametric test is appropriate given unknown distributions and modest sample sizes. Two-tailed p-values are reported with significance threshold $\alpha = 0.001$.

4. Results

4.1. Pre-trained Weight Spectral Analysis

We analyzed Query and Key projection matrices across all 12 transformer layers in mBERT and MuRIL. Singular value decomposition was performed on weight matrices of shape 768×768, and effective rank was computed using Equation 1.

Pre-trained attention weights exhibit mean effective ranks of 436.8 ± 33.5 (mBERT) and 441.0 ± 31.4 (MuRIL) across Query and Key matrices, with a combined average of 438.9 ± 32.5 ($n = 48$ matrices). Effective ranks range from 376.6 (mBERT Key, Layer 0) to 495.1 (MuRIL Query, Layer 3). The 90% energy rank averages 300.5 ± 26.9 (mBERT) and 302.5 ± 24.4 (MuRIL), with an overall mean of 301.5 ± 25.6, indicating that approximately 39% of dimensions (301/768) account for 90% of the total spectral energy (Table 2).

Effective ranks show no systematic monotonic trend across layers. The coefficient of variation is 7.7% for mBERT and 7.1% for MuRIL, indicating modest within-model variability. Query matrices have higher mean effective ranks than Key matrices in mBERT (448.6 vs. 425.1), whereas MuRIL exhibits near parity between Query and Key matrices (442.1 vs. 439.9). The difference in mean effective rank between models is small (4.2; 0.95% of the mean) relative to within-model variability, suggesting highly similar spectral structure despite differences in pre-training objectives and data (Figure 1).

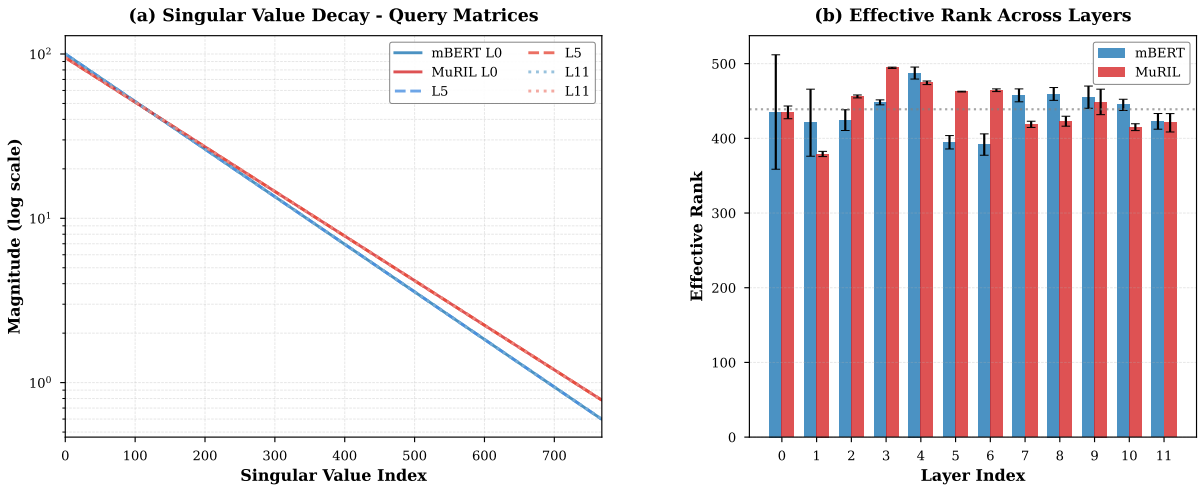


Figure 1: Spectral properties of pre-trained multilingual models. (Left) Singular value decay of Query projection matrices, illustrating smooth log-linear behavior indicative of high intrinsic dimensionality. (Right) Effective rank across transformer layers for mBERT and MuRIL, showing consistently high dimensionality with modest layer-wise variation.

4.2. Cross-Lingual Representation Geometry

We extracted sentence-level embeddings using mean pooling over token representations for 300 samples per language condition: (1) Hinglish (code-mixed), (2) Hindi-only, and (3) English-only. Embeddings were obtained from the final hidden states of mBERT and MuRIL. Cross-lingual similarity was quantified using CKA (Equation 3).

Hinglish embeddings exhibit strong alignment with Hindi, with CKA scores of 0.267 (mBERT) and 0.291 (MuRIL), yielding a mean of 0.279. Alignment between Hinglish and English embeddings is substantially weaker, with CKA values of 0.107 (mBERT) and 0.080 (MuRIL), yielding a mean of 0.093. The Hindi–English monolingual baseline yields lower similarity values (0.084 for mBERT and 0.065 for MuRIL; mean 0.074), as summarized in Table 3.

Hinglish–Hindi alignment is approximately 3.77× stronger than the Hindi–English baseline and about 3.0× stronger than Hinglish–English alignment, indicating asymmetric cross-lingual overlap induced by code-mixing. A permutation test (1,000 permutations) that disrupts correspondence between Hinglish and Hindi embeddings yields markedly lower CKA values (0.057 for mBERT and 0.046 for MuRIL), with $p < 0.001$ for the difference between original and permuted conditions. This confirms that the observed alignment reflects structured cross-lingual geometry rather than random similarity (Figure 2, Table 4).

Table 3: Cross-lingual representation similarity measured via CKA for mBERT and MuRIL. Bootstrap standard deviations shown.

Model	Pair	CKA	Standard Deviation
mBERT	Hinglish–Hindi	0.267	0.040
mBERT	Hinglish–English	0.107	0.012
mBERT	Hindi–English	0.084	0.011
MuRIL	Hinglish–Hindi	0.291	0.043
MuRIL	Hinglish–English	0.080	0.011
MuRIL	Hindi–English	0.065	0.009

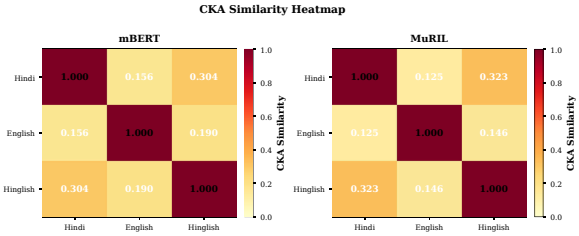


Figure 2: Cross-lingual representation similarity between Hinglish, Hindi, and English embeddings for mBERT and MuRIL, measured using CKA.

Table 4: Permutation ablation: Hinglish–Hindi CKA vs. permuted baseline ($p < 0.001$).

Condition	**mBERT**	**MuRIL**
Hinglish–Hindi (original)	0.267	0.291
Hinglish–Hindi (permuted)	0.057	0.046

Table 5: Token-level LID performance and parameter overhead for LoRA adapters.

r	Macro F1	Weighted F1	Trainable Params
4	0.9365	0.9400	223,491
8	0.9405	0.9438	444,675
16	0.9484	0.9513	887,043

Table 6: Effective ranks of pre-trained attention weights and LoRA update matrices.

r	Pre-trained Eff. Rank	LoRA Eff. Rank	Compression	90% Rank
4	441.6	2.11 ± 0.58	209×	1.6
8	441.6	3.56 ± 1.33	124×	2.6
16	441.6	5.92 ± 2.69	75×	3.8

4.3. LoRA Adaptation: Performance and Spectral Properties

LoRA adapters were trained end-to-end on token-level language identification using the COMI-LINGUA dataset (1,332 training, 166 development, and 168 test samples). Target modules were Query, Key, and Value projections. Adapter ranks $r \in \{4, 8, 16\}$ were evaluated using a scaling factor $\alpha = 2r$.

Test-set performance improves modestly with rank, from macro F1 of 0.937 ($r = 4$) to 0.948 ($r = 16$) which is a gain of only 1.2 percentage points despite doubling trainable parameters from 0.126% to 0.498% of the base model (Table 5, Figure 4).

Learned LoRA matrices were combined into effective update matrices $\Delta W = 2BA$ incorporating inference-time scaling. Mean effective ranks of LoRA updates are 2.11 ± 0.58 ($r = 4$), 3.56 ± 1.33 ($r = 8$), and 5.92 ± 2.69 ($r = 16$), corresponding to compression factors of 209.3×, 124.1×, and 74.6× relative to pre-trained weights (Table 6, Figure 3). A Wilcoxon signed-rank test comparing pre-trained effective ranks ($n = 48$) to LoRA effective ranks ($n = 108$) yields $p = 9.3 \times 10^{-20}$, rejecting the null hypothesis of equal ranks at $\alpha = 0.001$.

5. Discussion

5.1. From Multilingual Pre-training to Low-Rank Sufficiency

Our results provide convergent evidence linking multilingual representation geometry to the empirical efficiency of low-rank adaptation. Spectral analysis shows that pre-trained multilingual encoders allocate a high intrinsic dimensionality to linguistic knowledge, with mean effective ranks of approximately 439 across Query and Key matrices. This reflects the representational demands of encoding over one hundred languages within a shared parameter space.

In contrast, Hindi–English code-mixed text does not appear to require access to this full capacity. Cross-lingual geometry analysis reveals that Hinglish embeddings exhibit substantially stronger alignment with Hindi than with English, with Hinglish–Hindi similarity exceeding the Hindi–English monolingual baseline by a factor of 3.77. This asymmetric alignment indicates that code-mixed representations largely occupy an existing Hindi-dominated subspace rather than forming an independent intermediate space between languages. Permutation-based ablation further supports this interpretation: shuffling sample correspondence reduces Hinglish–Hindi similarity to near-zero levels, consistent with random pairings.

Given this geometric constraint, LoRA fine-tuning can achieve efficient adaptation by operating within a low-dimensional subspace that is already well supported by pre-training. Empirically, learned LoRA updates exhibit effective ranks between 2.1 and 5.9, corresponding to over two orders of magnitude compression relative to pre-trained attention weights, while maintaining strong task performance. The convergence between geometric overlap and empirical compression supports a causal interpretation: low-rank sufficiency arises because code-mixed representations are confined to a structured cross-lingual manifold rather than spanning the full representational space.

5.2. Asymmetric Cross-Lingual Alignment in Code-Mixing

The observed asymmetry between Hinglish–Hindi and Hinglish–English alignment is consistent with the matrix–embedded language structure of Hindi–English code-mixing. In the COMI-LINGUA corpus, Hindi typically serves as the matrix language providing syntactic structure, while English contributes embedded lexical material. As a result, Hindi subsequences extracted from code-mixed sentences retain contextual coherence, whereas English subsequences often reduce to isolated fragments.

This linguistic asymmetry propagates to representation geometry. Code-mixed embeddings preserve structural similarity to Hindi contexts, leading to stronger alignment with Hindi representations. In addition, multilingual pre-training exposes models to substantially more Hindi monolingual data than naturally occurring code-mixed text, resulting in a well-developed Hindi subspace into which Hinglish inputs naturally project. English representations, by contrast, are shaped primarily by monolingual English contexts that differ from embedded English usage in code-mixing, and therefore align

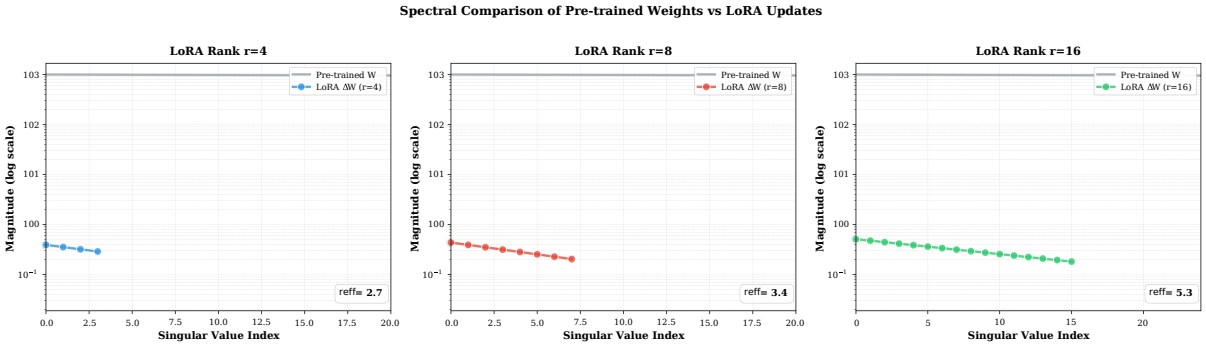


Figure 3: Spectral comparison of pre-trained attention weights and LoRA update matrices for $r = 4$ (left), $r = 8$ (center), and $r = 16$ (right), illustrating concentration of spectral energy in a low-dimensional subspace for LoRA updates.

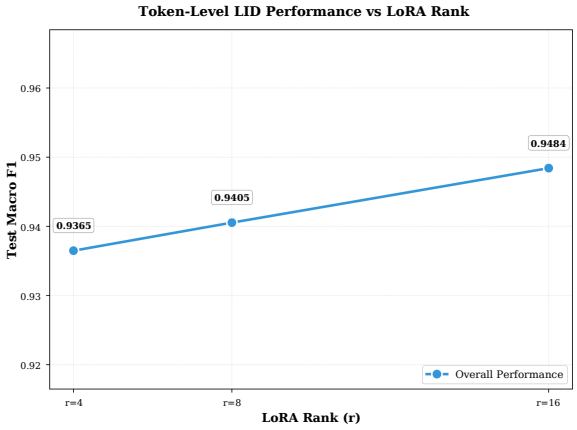


Figure 4: Token-level language identification performance as a function of LoRA rank, showing rapid saturation at low ranks.

less strongly.

We emphasize that this interpretation is specific to Hindi-matrix code-mixing. Scenarios in which English functions as the matrix language may exhibit reversed alignment patterns, and generalization to other language pairs requires empirical validation.

5.3. Implications for Parameter-Efficient Fine-Tuning

Our findings suggest that the effectiveness of LoRA for code-mixing stems from its ability to exploit pre-existing multilingual structure rather than from task-specific optimization alone. This implies that LoRA efficiency should vary systematically with cross-lingual alignment in pre-trained models: language pairs with stronger geometric overlap should admit lower-rank adaptation, while more distant pairs may require higher ranks or alternative methods.

More broadly, the analysis framework developed here linking pre-trained spectral properties, cross-lingual geometry, and adaptation rank may extend beyond code-mixing to other multilingual and domain-adaptation scenarios in which target distributions concentrate within structured subspaces of the pre-trained representation space.

6. Conclusion

We investigated why Low-Rank Adaptation enables efficient fine-tuning for code-mixed text through geometric and spectral analysis of multilingual models. Our results establish a clear link between cross-lingual representational structure and low-rank sufficiency: while pre-trained models exhibit high intrinsic dimensionality, Hindi–English code-mixed representations concentrate in a much lower dimensional subspace with substantially stronger alignment to Hindi than to English.

This geometric constraint enables LoRA adapters to achieve extreme compression while maintaining strong task performance. The convergence of spectral analysis, cross-lingual similarity measurements, and empirical adaptation results supports a unified interpretation: LoRA succeeds by refining pre-existing multilingual structure rather than learning new high-dimensional representations. Permutation-based ablations confirm that the observed alignment reflects semantic structure rather than artifacts, and consistency across models suggests that this phenomenon arises from fundamental properties of multilingual pre-training when Hindi serves as the matrix language.

Our analysis is demonstrated for token-level language identification on Hindi–English code-mixing. Whether similar alignment compression relationships hold across other language pairs, tasks, and model architectures remains an open question. Nevertheless, the framework presented here offers a mechanistic perspective on parameter-efficient fine-tuning grounded in representational geometry, rather than purely empirical rank selection.

7. Ethics Statement

This work analyzes publicly available multilingual language models and datasets to study parameter-efficient fine-tuning behavior. The COMI-LINGUA dataset consists of anonymized, publicly released social media text and does not contain personally identifiable information. No new data involving human subjects was collected.

While improved modeling of code-mixed language may benefit multilingual NLP applications, such systems may still reflect biases present in pre-training data. We therefore caution against deployment without appropriate evaluation of fairness, bias, and societal impact in downstream settings.

8. Limitations

Language Pair and Mixing Direction. Our analysis is restricted to Hindi–English code-mixing where Hindi serves as the matrix language. Other language pairs or mixing directions may exhibit different representational geometry and adaptation behavior. Findings are most likely to generalize to other Indo-Aryan pairs where one language dominates as the matrix language (e.g., Hindi–Urdu, Hindi–Marathi), though empirical validation is required.

Task Scope. We focus on token-level language identification, a relatively constrained classification task. More complex sentence-level or generative tasks may require higher-dimensional adaptations than those observed here. Token-level language identification was selected as it provides clean, interpretable geometric signal with direct per-token ground-truth labels. Other applicable scenarios include multilingual domain adaptation and cross-lingual transfer between typologically similar languages.

Dataset Size. The filtered dataset contains 1,666 samples, limiting statistical power for detecting small effects and for analyzing rare mixing patterns. Larger datasets would enable more fine-grained analysis. However, CKA and spectral decomposition operate on embedding matrices rather than trained classifiers, and bootstrap confidence intervals confirm stability of the similarity estimates.

Correlation vs. Causation. Although we present convergent evidence linking cross-lingual alignment to low-rank adaptation efficiency, the analysis remains correlational. We do not perform interventional experiments that explicitly manipulate representation geometry.

Model Coverage. Experiments are limited to two encoder-only multilingual models. Extending the analysis to larger or decoder-only architectures may reveal additional factors influencing low-rank adaptation behavior. XLM-R and LaBSE are natural candidates for future analysis; LaBSE's alignment-optimized pre-training may exhibit higher CKA values, which our framework predicts would correlate with lower LoRA ranks. Extending this analysis to decoder-only architectures such as Llama or Mistral, and to billion-parameter models, remains open.

9. Acknowledgements

This work was carried out using publicly available pre-trained models and open-source libraries, including PyTorch, HuggingFace Transformers, and PEFT. Large language models were used solely for language polishing and final editorial refinement of the manuscript. All scientific content, experiments, and conclusions are the authors' own.

10. Data and Code Availability

The code for all experiments, including spectral analysis, CKA computation, and LoRA fine-tuning, is available at `https://github.com/shakuvish/LoRA-Geometric`. The COMI-LINGUA dataset used in this study is publicly available via Sheth et al. (2025).

11. Bibliographical References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin,

Ireland. Association for Computational Linguistics.

Dan Biderman, Jose Javier Gonzalez Ortiz, Ranganath Jacob, Matthew Sullivan, Sainbayar Sukhbaatar, Martin Wistuba, et al. 2024. LoRA learns less and forgets less. *arXiv preprint arXiv:2405.09673*.

John Dang, Arash Ahmadian, Kelly Marchisio, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. 2024. Rlhf can speak many languages: Unlocking multilingual preference optimization for llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13134–13156, Miami, Florida, USA. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280, Online. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Agrawal, Pooja Hase, Pushpak Gupta, Palak Singh, et al. 2021. MuRIL: Multilingual representations for Indian languages. *arXiv preprint arXiv:2103.10730*.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3519–3529. PMLR.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. 2023. A kernel-based view of language model fine-tuning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 23610–23641. PMLR.

Charles H. Martin and Michael W. Mahoney. 2019. Traditional and heavy-tailed self regularization in neural network models. In *arXiv preprint*, volume abs/1901.08276.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, volume 31.

Pieter Muysken. 2000. *Bilingual Speech: A Typology of Code-Mixing*. Cambridge University Press, Cambridge.

Carol Myers-Scotton. 1993. *Duelling Languages: Grammatical Structure in Codeswitching*. Oxford University Press, Oxford.

Minh Van Nguyen, Tuan Ngo Nguyen, Bonan Min, and Thien Huu Nguyen. 2021. Crosslingual transfer learning for relation and event extraction via word category and class alignments. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5414–5426, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards subword level compositions for sentiment analysis of Hindi-English code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491, Osaka, Japan. The COLING 2016 Organizing Committee.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on Twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1971–1982, Vancouver, Canada. Association for Computational Linguistics.

Kaushal Sharma, Ajit Kembhavi, Aniruddha Kembhavi, T Sivarani, Sheelu Abraham, and Kaustubh Vaghmare. 2020. Application of convolutional neural networks for stellar spectral classification. *Monthly Notices of the Royal Astronomical Society*, 491(2):2280–2300.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018. Language identification and named entity recognition in Hinglish code mixed tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 52–58, Melbourne, Australia. Association for Computational Linguistics.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

12. Language Resource References

Sheth, Rajvee and Beniwal, Himanshu and Singh, Mayank. 2025. *COMI-LINGUA: Expert Annotated Large-Scale Dataset for Multitask NLP in Hindi–English Code-Mixing*. Association for Computational Linguistics. PID https://aclanthology.org/2025.findings-emnlp.422/.

Hi-SEMFLOW: Lie Algebra–Based Semantic Flow for Span-Level Informal Language Identification in Hindi

Manikandan Ravikiran[1*], Tanmay Tiwari[2*], Vibhu Gupta[2], Rohit Saluja[2,3]
Thoughtworks AI Labs[1]
Indian Institute of Technology Mandi[2]
BharatGen Consortium[3]
manikandan.r@thoughtworks.com
{s23107,b22248}@students.iitmandi.ac.in
rohit@iitmandi.ac.in

Abstract

Informal Hindi text frequently contains multi-token slang and idiomatic expressions whose correct identification requires consistent span boundaries. Transformer-based token classifiers, despite strong contextual representations, often produce fragmented or structurally invalid BIO sequences due to largely local predictions. We propose Hi-SEMFLOW, a Lie algebra–based semantic flow framework that models span consistency as a continuous refinement process over label logits. Instead of discrete structured decoding, Hi-SEMFLOW learns context-dependent transition operators derived from antisymmetric generators and propagates structural information through smooth, fully differentiable transformations. This formulation integrates structural bias directly into end-to-end training without requiring dynamic programming or hard decoding constraints. Experiments on the HiSlang-4.9k benchmark show that Hi-SEMFLOW improves span-level F1 by up to 2–3 absolute points and yields consistent macro-F1 gains across Hindi-pretrained encoders. Extensive ablations demonstrate that continuous geometric refinement provides a flexible and effective alternative to discrete structured decoding for span-centric sequence labeling.

Keywords: informal language identification, structured sequence labeling, geometric label propagation

1. Introduction

Informal language is pervasive in contemporary online communication, particularly across social media, messaging platforms, and conversational digital content (Mustofa and Saptomo, 2025). Despite being spoken by over 600 million people worldwide, Hindi remains significantly underrepresented in NLP research, accounting for less than 1% of web content (Q-Success, 2024). This imbalance is especially pronounced for informal digital text, where linguistic variation is high and annotated resources are scarce. Hindi informal language frequently includes slang expressions, colloquial constructions, phonetic spellings, and non-standard orthography that diverge from canonical usage (Tiwari et al., 2025). These phenomena challenge downstream NLP systems such as parsing, sentiment analysis, and machine translation, motivating the need for robust informal language identification methods (Pei et al., 2019).

In this work, we study token-level informal language identification for Hindi, formulated as a BIO-style sequence labeling task. The objective is to detect contiguous spans of informal expressions within a sentence. Unlike isolated lexical anomalies, informal expressions often span multiple tokens and exhibit flexible boundaries. As a result, the task is inherently *structural*: correct predictions require not only identifying informal tokens, but also maintaining consistent span boundaries across the sequence. Fragmented spans or invalid BIO transitions (e.g., O → I-INF) directly degrade span-level performance. Most existing approaches employ transformer-based encoders with independent token classification (Sun et al., 2024; Tiwari et al., 2025). While contextualized representations are powerful, predictions remain largely local. This often leads to boundary inconsistencies, such as prematurely terminated spans or isolated continuation tags (Ravikiran et al., 2026). A common solution is to incorporate conditional random fields (CRFs) or constrained decoding to enforce BIO legality (Lafferty et al., 2001; Lester et al., 2020). However, these methods rely on discrete structured inference, introduce additional decoding complexity, and tightly couple training to a specific inference procedure. Moreover, hard decoding constraints may be overly rigid in settings where span boundaries are ambiguous or context-dependent.

Accordingly, we propose Hi-SEMFLOW (Hindi Semantic Flow), a fully differentiable structural refinement framework that models span consistency as a *continuous transformation of label logits*. Rather than imposing hard transition constraints through discrete decoding, Hi-SEMFLOW performs smooth, learnable refinement in logit space us-

*Authors contributed equally to this work. Names are listed in alphabetical order.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 137–153
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

ing a Lie algebra–based semantic flow module. Specifically, the model parameterizes context-dependent transition operators via antisymmetric generators and applies them as position-sensitive transformations across adjacent tokens. Unlike CRFs, which rely on fixed global transition matrices and dynamic programming, `Hi-SEMFLOW` learns adaptive local transition dynamics that softly encourage valid BIO sequences (e.g., `B-INF` → `I-INF`) while preserving strong lexical evidence. Structural bias is thus integrated directly into end-to-end training rather than enforced through post-hoc decoding constraints. We evaluate `Hi-SEMFLOW` on the `HiSlang-4.9k` benchmark (Tiwari et al., 2025) using both Hindi-specific and multilingual pretrained encoders. Across settings, semantic flow reduces invalid BIO transitions by up to 50%, improves span-level F1 by 2–3 absolute points, and yields consistent macro-F1 gains for Hindi-pretrained models. Ablation studies further analyze the effect of generator size, refinement strength, and interaction with decoding constraints, highlighting the importance of calibrated structural propagation. In summary, our contributions are:

- We introduce `Hi-SEMFLOW`, a fully differentiable semantic flow framework that models BIO transition dynamics as continuous geometric refinement over label logits.
- We provide comprehensive empirical evaluation on `HiSlang-4.9k`, demonstrating consistent span-level improvements and substantial reductions in invalid BIO transitions across Hindi-specific and multilingual encoders.

2. Related Work

Informal Language Identification in Hindi Informal language identification has gained attention due to the prevalence of non-standard language in social media and conversational text (Eisenstein, 2013). In Hindi and other Indic languages, informal usage includes slang expressions, phonetic spellings, and orthographic variation (Vyas et al., 2014). Recent work such as `HiSlang-4.9k`(Tiwari et al., 2025) formulates slang detection as a token-level sequence labeling task using BIO annotations (Ramshaw and Marcus, 1995; Murthy et al., 2022). While transformer-based models achieve strong token-level performance, predicted spans often exhibit fragmentation and boundary inconsistencies, especially for multi-token informal expressions.

Transformer-Based Sequence Labeling Transformer encoders such as BERT (Devlin et al., 2019) and its multilingual and Indic variants including mBERT, IndicBERT (Kakwani et al., 2020), XLM-RoBERTa (Conneau et al., 2020), and MuRIL (Khanuja et al., 2021) are now standard for sequence labeling tasks, including NER and span detection (Keraghel et al., 2024). These models generate contextual token representations followed by independent classification at each position. Although self-attention captures contextual information, explicit span-level inductive bias is typically absent, and structural dependencies are only weakly enforced during training (Tedeschi et al., 2022).

CRFs and BIO Constraints Linear-chain conditional random fields (CRFs) (Lafferty et al., 2001) are commonly used to impose BIO consistency. Neural architectures such as BiLSTM-CRF (Ma and Hovy, 2016) and transformer-CRF hybrids enforce label transitions through discrete dynamic programming. Alternative approaches apply constrained decoding or masking to ensure BIO legality (Lester et al., 2020; Ratinov and Roth, 2009). However, these methods treat structural consistency as a hard decoding constraint rather than as a learnable property of the model (Archana et al., 2023). This separation can limit flexibility, particularly in settings where span boundaries are ambiguous (Singh et al., 2018).

Soft Structured Prediction and Geometry-Inspired Methods To model structured dependencies without discrete inference, prior work has explored differentiable formulations of structured prediction. Conditional random fields have been reformulated as recurrent networks, unrolling inference as differentiable layers (Zheng et al., 2015). Energy-based structured models, such as Structured Prediction Energy Networks (SPENs), similarly integrate label interactions into end-to-end optimization without relying on dynamic programming (Belanger and McCallum, 2016). Other approaches employ soft or iterative label refinement mechanisms that propagate prediction information across positions while preserving differentiability (Cui and Zhang, 2019). These methods regularize label interactions through continuous updates rather than discrete combinatorial decoding. In parallel, geometric deep learning introduces architectures grounded in Lie group theory and symmetry principles (Cohen and Welling, 2016; Bronstein et al., 2021), and Lie algebra parameterizations have been used to model smooth transformations in vision, robotics, and representation learning (Gerken et al., 2023; Falorsi et al., 2019). However, Lie-theoretic constructions remain largely unexplored in NLP structured prediction. Unlike equivariant models that constrain representation space, we parameterize transition

dynamics directly in label space via antisymmetric generators and exponential maps, modeling BIO dependencies as smooth geometric flows. This enables position-specific structural propagation rather than global transition scoring, providing adaptive, context-conditioned structural bias while preserving full end-to-end differentiability.

3. Methodology

We present `Hi-SEMFLOW` as a differentiable structural refinement layer built on top of a transformer-based token classifier. Instead of imposing discrete transition constraints during decoding, the method constructs position-specific transition operators in label space and applies them sequentially to propagate structural information across adjacent tokens. Algorithm 1 summarizes the forward pass. Starting from encoder-derived token logits, we (i) compute context-conditioned transition operators via antisymmetric generator composition, (ii) propagate structural evidence from position $t-1$ to t through a matrix exponential mapping, and (iii) interpolate between local lexical evidence and structurally propagated logits. Training integrates a curriculum on structural strength together with geometric regularization to ensure stable and well-conditioned refinement. More detailed presentation in Appendix A.

Algorithm 1 `Hi-SEMFLOW` (Forward Pass)

Require: Token sequence $X = (x_1, \dots, x_T)$; encoder Encoder$(\cdot)$; classifier parameters (W, b); generator matrices $\{G_k\}_{k=1}^{K}$; projection head f_θ; refinement strength λ
Ensure: Refined probabilities $\{\hat{p}_t\}_{t=1}^{T}$

for $t \leftarrow 1$ to T **do**
  $h_t \leftarrow \text{Encoder}(X)_t$
  $z_t \leftarrow W h_t + b$
end for
for $k \leftarrow 1$ to K **do**
  $\tilde{G}_k \leftarrow \frac{1}{2}(G_k - G_k^\top)$
end for
$\hat{z}_1 \leftarrow z_1$
for $t \leftarrow 2$ to T **do**
  $\alpha_t \leftarrow f_\theta(h_t)$
  $A_t \leftarrow \sum_{k=1}^{K} \alpha_{t,k} \tilde{G}_k$
  $F_t \leftarrow \exp(A_t)$ ▷ 3rd-order Taylor approximation
  $\tilde{z}_t \leftarrow F_t z_{t-1}$
  $\hat{z}_t \leftarrow (1-\lambda) z_t + \lambda \tilde{z}_t$
end for
for $t \leftarrow 1$ to T **do**
  $\hat{p}_t \leftarrow \text{softmax}(\hat{z}_t)$
end for
return $\{\hat{p}_t\}_{t=1}^{T}$

3.1. Flow-Based Structural Refinement

We describe the key components of Algorithm 1.

Base Token Predictions. Given an input sequence $X = (x_1, \dots, x_T)$, a pretrained encoder produces contextual representations

$$h_t = \text{Encoder}(X)_t \in \mathbb{R}^d,$$

which are mapped to token-level label logits

$$z_t = W h_t + b \in \mathbb{R}^{|\mathcal{L}|},$$

where $\mathcal{L} = \{\text{B-INF}, \text{I-INF}, \text{O}\}$. These logits represent independent token-level confidence over BIO labels.

Context-Conditioned Transition Operators. To model structured dependencies across adjacent tokens, we learn K generator matrices

$$G_k \in \mathbb{R}^{|\mathcal{L}| \times |\mathcal{L}|}.$$

Each matrix is antisymmetrized:

$$\tilde{G}_k = \frac{1}{2}(G_k - G_k^\top),$$

which yields approximately norm-preserving transformations after exponentiation and prevents uncontrolled logit amplification.

For each position t, a projection head predicts coefficients

$$\alpha_t = f_\theta(h_t) \in \mathbb{R}^K,$$

which combine generators into a position-specific operator

$$A_t = \sum_{k=1}^{K} \alpha_{t,k} \tilde{G}_k.$$

The transition matrix is obtained via

$$F_t = \exp(A_t).$$

Unlike CRFs that use a single global transition matrix, F_t is conditioned on contextual representations and varies across positions. The antisymmetric generator parameterization constrains transitions to well-conditioned geometric flows rather than unconstrained linear transformations. In practice, we approximate $\exp(A_t)$ using a third-order Taylor expansion, which provided a stable and efficient trade-off in experiments.

Logit-Space Refinement. Structural propagation transforms the previous token's logits:

$$\tilde{z}_t = F_t z_{t-1}.$$

We propagate from the base logits z_{t-1} rather than recursively refined logits to avoid compounding structural bias across long spans and to preserve strong lexical evidence. The refined logits interpolate between local and propagated evidence:

$$\hat{z}_t = (1-\lambda) z_t + \lambda \tilde{z}_t,$$

with final probabilities

$$\hat{p}_t = \text{softmax}(\hat{z}_t).$$

The refinement strength λ controls the balance between lexical evidence and structural consistency. We apply refinement left-to-right to align with directional BIO dependencies.

3.2. Stabilization and Training Control

Strong structural influence early in training may hinder lexical learning. We therefore adopt a curriculum schedule that gradually increases structural strength:

$$\lambda^{(e)} = \alpha_{\text{min}} + \frac{e}{E}(\alpha_{\text{max}} - \alpha_{\text{min}}),$$

where e is the current epoch and E is the total number of epochs.

To stabilize transition dynamics, we introduce lightweight geometric regularization:

Magnitude Control

$$\mathcal{L}_{\text{mag}} = \frac{1}{T}\sum_t \|\alpha_t\|_2^2,$$

Temporal Smoothness

$$\mathcal{L}_{\text{temp}} = \frac{1}{T-1}\sum_t \|\alpha_t - \alpha_{t+1}\|_2^2,$$

Orthogonality Preservation

$$\mathcal{L}_{\text{ortho}} = \frac{1}{T}\sum_t \|F_t^\top F_t - I\|_F^2.$$

The total flow regularization is

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{mag}} + \beta\mathcal{L}_{\text{ortho}} + \gamma\mathcal{L}_{\text{temp}}.$$

We found that combining these lightweight constraints yields stable training without architectural modification.

3.3. Optional Structural Constraints

To disentangle the effect of continuous refinement from conventional legality enforcement, we optionally apply decoding-time BIO masking:

$$z_t^{\text{BIO}} = z_t + \log(\text{softmax}(z_{t-1})M),$$

where M encodes valid BIO transitions. This allows evaluation of semantic flow independently or in combination with discrete decoding constraints.

3.4. Training Objective and Complexity

The final objective combines token-level cross-entropy with flow regularization:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \eta\mathcal{L}_{\text{flow}}.$$

Matrix exponentiation incurs $O(|\mathcal{L}|^3)$ cost per token. Since $|\mathcal{L}| = 3$ in our task, the overhead is negligible relative to encoder computation. For moderate label sizes, the complexity remains practical for sequence labeling.

4. Experimental Setup

4.1. Dataset

We evaluate on `HiSlang-4.9k` (Tiwari et al., 2025), a publicly available Hindi benchmark annotated for informal expressions. The dataset contains 4,906 sentences, evenly split between sentences containing at least one annotated expression (2,453) and sentences containing none (2,453). Data are drawn from movie scripts, subtitles, linguistic corpora, and social media, covering diverse writing styles. Although labeled as *slang*, the annotations encompass a broader class of informal expressions, including slang, idioms, and lexicalized multi-word phrases that function as semantically atomic units in context. Phrase-level spans are annotated using BIO tags, enabling sequence labeling with high inter-annotator agreement (Cohen's $\kappa = 0.97$ at the sentence level and $\kappa = 0.94$ at the phrase level). Importantly, many words appearing inside annotated expressions also occur elsewhere in literal contexts, preventing trivial word-based heuristics and requiring span-level modeling. We follow the official 80:20 train-test split. From the training portion, we hold out 10% as a development set.

4.2. Evaluation Metrics

We evaluate performance using token-level Precision (P), Recall (R), and macro-averaged F1-score, following the protocol of `HiSlang-4.9k` (Tiwari et al., 2025). Macro-F1 is reported as the primary metric to ensure balanced evaluation across BIO labels $\{\text{B-INF}, \text{I-INF}, \text{O}\}$ and to mitigate dominance of the majority `O` class. (More details in Appendix C)

Span-Level Evaluation. Because informal expressions frequently span multiple contiguous tokens, we additionally report span-level F1-score computed using exact span matching. A predicted span is counted as correct only if its boundaries and label type exactly match the gold annotation. This metric directly evaluates phrase-level coherence rather than isolated token accuracy.

BIO Validity Rate. To quantify structural consistency, we compute the invalid transition rate, defined as the percentage of adjacent label transitions that violate BIO legality constraints (e.g., `O` → `I-INF` without a preceding `B-INF`). Lower values indicate stronger structural coherence and reduced span fragmentation.

4.3. Implementation Details

Encoders. We evaluate the semantic flow module across diverse pretrained transformer encoders to assess generality: (a) `ai4bharat/IndicBERTv2-MLM-Sam-TLM` (IndicBERTv2), (b) `ai4bharat/indic-bert` (IndicBERT), (c) `google/muril-base-cased` (MuRIL), (d) `bert-base-multilingual-cased` (mBERT), (e) `xlm-roberta-base` (XLM-R), and (f) `bert-base-cased` (BERT). All encoders are fine-tuned end-to-end for sequence labeling.

Unless otherwise specified, we use $K = 8$ generators in the semantic flow module. Generator matrices are randomly initialized and antisymmetrized as described in Section 3. The exponential map is approximated using a third-order Taylor expansion. The refinement strength λ follows a linear curriculum schedule from $\lambda_{\text{min}} = 0.1$ to $\lambda_{\text{max}} = 0.5$ across training epochs. Alternative values are explored in ablations. Flow regularization weights are set to $\eta = 0.1$, $\beta = 0.01$, and $\gamma = 0.01$, selected based on development-set performance.

Models are trained using AdamW with learning rate 2×10^{-5}, batch size 32, and weight decay 0.01. Maximum sequence length is 128. Training runs for up to 10 epochs with early stopping based on development macro-F1. No additional learning rate scheduler is applied. Dropout is set to 0.1 (inherited from the pretrained encoder configuration). Gradient clipping is applied with max norm 1.0. In selected ablations, flow parameters are frozen after the first 3 epochs to examine their role as adaptive structural refiners.

5. Main Results

We evaluate whether Lie algebra–based semantic flow improves structured span prediction beyond standard transformer-based token classification. While macro-F1 provides a general performance signal, our primary focus is structural consistency, measured through invalid BIO transitions and span-level F1. Table 1 compares macro-averaged Precision, Recall, and F1 between baseline models and their `Hi-SEMFLOW` counterparts. `Hi-SEMFLOW` improves macro-F1 for most encoders, with particularly consistent gains for Hindi-pretrained models. MuRIL improves from 0.9332 to 0.9414 (+0.82), IndicBERTv2 from 0.9309 to 0.9373 (+0.64), and IndicBERT from 0.8486 to 0.8688 (+2.02). Moderate improvements are also observed for mBERT (0.9079 → 0.9205) and BERT-base (0.8624 → 0.8709). In contrast, XLM-R shows a slight decrease (0.9346 → 0.9307), suggesting that high-capacity multilingual encoders may already encode strong contextual dependencies, making additional structural refinement less beneficial under the same configuration. These results indicate that semantic flow interacts with encoder inductive bias and capacity, providing the most consistent benefits when contextual representations lack explicit span-level regularization.

Table 2 reports invalid BIO transition rates and exact span-level F1. Semantic flow substantially reduces structural violations across encoders. For IndicBERT, invalid BIO transitions decrease from 3.8% to 1.9% (50% relative reduction). MuRIL reduces from 2.6% to 1.2%, and IndicBERTv2 from 2.4% to 1.3%. Similar trends hold for other models. Importantly, span-level F1 improvements are consistent even when macro-F1 gains are modest. Indic-BERT improves from 0.821 to 0.844 (+2.3 points), MuRIL from 0.905 to 0.918 (+1.3), and IndicBERTv2 from 0.931 to 0.936 (+0.5). Smaller but positive gains are observed for mBERT, XLM-R, and BERT-base. These findings confirm that semantic flow primarily improves boundary coherence and reduces span fragmentation, rather than merely adjusting isolated token predictions.

6. Ablation Studies

We analyze how individual design choices in the semantic flow module influence performance. Unless otherwise specified, results are reported using macro-F1. The main experiments use $K = 8$ generators, which provides stable performance across most encoders.

Effect of Number of Generators We vary the number of Lie algebra generators K while keeping all other parameters fixed (Table 3). Performance sensitivity to K is model-dependent. IndicBERTv2 and XLM-R achieve their highest scores at $K = 8$, mBERT also peaks at $K = 8$, while MuRIL performs best at $K = 16$. In contrast, IndicBERT shows degradation as K increases beyond 2. Across encoders, excessively large generator sets ($K = 32$) consistently reduce performance. This may indicate over-parameterization or unstable structural refinement when structural expressiveness becomes too large relative to the label space ($|L| = 3$). Overall, moderate generator sizes ($K \in [4, 16]$) provide the most reliable behavior.

Model	Baselines (Tiwari et al., 2025)			Hi-SEMFLOW (Ours)		
	P	R	F1	P	R	F1
MuRIL	0.9348	0.9318	0.9332	**0.9413**	**0.9416**	**0.9414**
IndicBERTv2	0.9303	0.9317	0.9309	**0.9415**	**0.9335**	**0.9373**
XLM-R	**0.9342**	**0.9351**	**0.9346**	0.9315	0.9300	0.9307
mBERT	0.9034	0.9125	0.9079	**0.9196**	**0.9215**	**0.9205**
BERT	0.8704	0.8547	0.8624	**0.8875**	**0.8557**	**0.8709**
IndicBERT	0.8524	0.8450	0.8486	**0.8780**	**0.8603**	**0.8688**

Table 1: Comparison of models with Hi-SEMFLOW on HiSlang-4.9k Benchmark.

Model	Invalid BIO (%) ↓	Span-F1 ↑
IndicBERT	3.8 → 1.9	0.821 → 0.844
IndicBERTv2	2.4 → 1.3	0.931 → 0.936
MuRIL	2.6 → 1.2	0.905 → 0.918
mBERT	2.1 → 1.5	0.908 → 0.912
XLM-R	2.3 → 1.6	0.934 → 0.938
BERT	2.9 → 2.0	0.862 → 0.864

Table 2: Effect of semantic flow on structural validity and span-level performance across all evaluated encoders. Arrows indicate performance change from No Flow to Flow.

Model	2	4	8	16	32
IndicBERTv2	0.9340	0.9334	0.9348	0.9342	0.9268
IndicBERT	0.8596	0.8515	0.8433	0.8510	0.8529
BERT-base	0.8604	0.8626	0.8547	0.8630	0.8614
mBERT	0.9149	0.9046	0.9194	0.9079	0.9166
MuRIL	0.9342	0.9333	0.9309	0.9410	0.9304
XLM-R	0.9251	0.9283	0.9296	0.9263	0.9256

Table 3: Macro-F1 across different numbers of generators K.

Effect of Curriculum Strength We evaluate different ranges for flow strength λ (See Table 4). Curriculum schedules are implemented as linear increases of λ across training epochs with the following ranges: Weak = $[0.05, 0.3]$, Medium = $[0.1, 0.5]$, Strong = $[0.2, 0.7]$, Very Strong = $[0.3, 0.9]$.

The optimal schedule varies across encoders. IndicBERTv2 and XLM-R perform best under the medium configuration, mBERT peaks under the strong configuration, and MuRIL achieves its highest score under the weak schedule. However, very strong structural influence consistently degrades performance, particularly for IndicBERT and MuRIL. These results suggest that excessive structural refinement may induce over-smoothing. Moderate schedules provide more stable behavior across models.

Interaction with BIO Constraints We analyze the interaction between semantic flow and decoding-time BIO masking (See Table 5). Here, *None* denotes the baseline transformer classifier without flow or decoding constraints; *Flow* denotes semantic flow without BIO masking; *Decode* denotes decoding-time BIO masking without flow; and *Both* applies both mechanisms.

Model	Weak	Medium	Strong	Very Strong
IndicBERTv2	0.9317	0.9355	0.9323	0.9279
IndicBERT	0.8517	0.8488	0.8466	0.8307
BERT-base	0.8538	0.8619	0.8608	0.8530
mBERT	0.9120	0.9127	0.9153	0.9128
MuRIL	0.9350	0.9313	0.9276	0.9132
XLM-R	0.9267	0.9307	0.9189	0.9200

Table 4: Macro-F1 across curriculum strengths.

Model	None	Flow	Decode	Both
IndicBERTv2	0.9335	0.9296	0.9336	0.9305
IndicBERT	0.8526	0.8483	0.8409	0.8574
BERT-base	0.8664	0.8660	0.8597	0.8544
mBERT	0.9180	0.9085	0.9129	0.9205
MuRIL	0.9274	0.9295	0.9309	0.9297
XLM-R	0.9223	0.9264	0.9306	0.9285

Table 5: Macro-F1 for different structural enforcement mechanisms.

The interaction between continuous flow and discrete decoding constraints varies across encoders. For mBERT, combining both mechanisms yields the highest F1 (0.9205). IndicBERT also benefits from the combined configuration. However, for IndicBERTv2 and MuRIL, decode-only or baseline settings perform comparably to flow-based refinement. These results suggest partial complementarity: semantic flow can approximate structured decoding in some settings, but its interaction with discrete masking depends on encoder characteristics.

Effect of Flow Freezing We examine freezing flow parameters after different epochs (See Table 6). For this experiment, training was extended to 50 epochs to allow evaluation of late freezing effects; freeze epoch denotes the epoch at which flow parameters are fixed for the remainder of training.

The effect of freezing is encoder-dependent. IndicBERTv2 benefits from late freezing (epoch 40), whereas Indic-BERT performs best with early freezing (epoch 10). MuRIL and mBERT achieve their strongest results without freezing. These pat-

Model	Never	10	20	30	40
IndicBERTv2	0.9314	0.9324	0.9325	0.9287	0.9373
Indic-BERT	0.8673	0.8688	0.8515	0.8592	0.8470
BERT-base	0.8659	0.8709	0.8674	0.8599	0.8589
mBERT	0.9187	0.9083	0.9159	0.9150	0.9060
MuRIL	0.9414	0.9266	0.9359	0.9317	0.9380
XLM-R	0.9282	0.9288	0.9302	0.9192	0.9277

Table 6: Macro-F1 when freezing flow at different epochs.

terns indicate that semantic flow acts as a structural regularizer whose optimal training duration varies across models.

Generators $\times$ BIO Interaction To examine whether continuous structural refinement and discrete BIO masking are complementary or redundant, we compare flow-only configurations at $K = 8$ (the default setting) with decode-only and combined settings (Table 7).

Model	Flow ($K = 8$)	Decode	Both
IndicBERT	0.8433	0.8409	0.8574
mBERT	0.9194	0.9129	0.9205
XLM-R	0.9296	0.9306	0.9285

Table 7: Interaction between generator size ($K = 8$) and BIO masking (macro-F1).

For IndicBERT, flow-only slightly outperforms decode-only (0.8433 vs. 0.8409), while combining both mechanisms yields the highest score (0.8574), indicating complementary behavior in moderate-capacity encoders. For mBERT, flow-only already improves over decode-only (0.9194 vs. 0.9129), and the combined configuration produces a marginal additional gain (0.9205), suggesting partial complementarity. In contrast, for XLM-R, decode-only slightly exceeds flow-only (0.9306 vs. 0.9296), and combining both does not provide further improvement (0.9285). This indicates potential redundancy when strong contextual modeling already captures transition dynamics. Taken together, these results suggest that moderate generator sizes provide sufficient structural capacity, and that combining continuous refinement with discrete masking can be beneficial in some encoders but may introduce redundancy in high-capacity multilingual models. Structural expressiveness therefore requires calibration rather than maximal constraint stacking.

7. Analysis

We analyze when and why semantic flow improves span prediction, and identify scenarios where structural refinement is most beneficial. Additional quantitative analysis of boundary errors and structural validity is provided in Appendix E.

7.1. Span-Length Sensitivity

Informal expressions in `HiSlang-4.9k` frequently span multiple tokens. We therefore evaluate performance grouped by span length: single-token spans, short spans (2–3 tokens), and long spans (4+ tokens).

Across encoders, improvements are concentrated in multi-token spans. For Indic-BERT, overall span-F1 improves from 0.821 to 0.844 (+2.3), with gains primarily attributable to longer expressions such as काँटों पर चलने वाला हाल होना and खुशी का ठिकाना नहीं रहा which span four or more tokens. MuRIL exhibits a similar pattern, where span-F1 increases from 0.905 to 0.918 (+1.3), largely driven by improved boundary consistency in multi-token idiomatic phrases. In contrast, single-token slang expressions show minimal change across models. This pattern indicates that semantic flow primarily improves boundary coherence and internal span continuity rather than isolated token classification.

7.2. Boundary Error Analysis

Independent token classifiers commonly exhibit boundary inconsistencies, including (a) Premature termination of spans (predicting O within a multi-token span), (b) Missing initial B-SLANG labels, (c) Invalid O $\rightarrow$ I-SLANG transitions. For example, expressions such as हाथ पैर फूल जाना and खुद को लुटाना are occasionally predicted as partial spans by baseline models, labeling only the first token or prematurely reverting to O. As shown in Table 2, semantic flow reduces invalid BIO transitions (e.g., 3.8% $\rightarrow$ 1.9% for Indic-BERT; 2.6% $\rightarrow$ 1.2% for MuRIL). These reductions correspond to fewer fragmented predictions and improved span-level F1. Qualitative inspection indicates that flow-based refinement often converts inconsistent sequences such as B-SLANG O I-SLANG into coherent predictions B-SLANG I-SLANG I-SLANG, thereby restoring full span boundaries.

7.3. When Semantic Flow Degrades Performance

Although semantic flow improves structural validity in most configurations, slight degradation is observed in certain high-capacity models under strong refinement settings. For example, XLM-R shows reduced macro-F1 under strong or very strong curricula (Table 4), and performance declines for large generator counts ($K = 32$; Table 3). In such cases, excessive structural influence may over-smooth locally confident predictions, particularly for rare or context-specific expressions. For

instance, phrases embedded within otherwise formal sentences such as भारत रत्न मिलना चाहिए may occasionally be over-extended by one token when structural refinement is too strong. These observations indicate that semantic flow functions as a calibration mechanism whose effectiveness depends on appropriate tuning of generator size and curriculum strength.

7.4. Qualitative Examples

We analyze representative cases to understand how semantic flow alters token-level decisions and improves structural consistency.

Example 1: Internal Fragmentation Correction काँटों पर चलने वाला हाल होना (kāṭõ par calne vālā hāl honā; "to be in extreme hardship")

Token	Gold	Baseline	Hi-SEMFLOW
काँटों	B-SLANG	B-SLANG	B-SLANG
पर	I-SLANG	O	I-SLANG
चलने	I-SLANG	O	I-SLANG
वाला	I-SLANG	I-SLANG	I-SLANG
हाल	I-SLANG	O	I-SLANG
होना	I-SLANG	O	I-SLANG

Table 8: Correction of fragmented span under semantic flow.

The baseline model exhibits internal discontinuities, reverting to O despite strong contextual continuity. Semantic flow restores span integrity by propagating continuation probabilities forward, resulting in a structurally consistent sequence. This behavior aligns with the observed reduction in invalid BIO transitions and improved span-level F1.

Example 2: Boundary Continuity in Multi-Token Expressions खुशी का ठिकाना नहीं रहा (khuśī kā ṭhikānā nahī rahā; "there was no limit to the happiness")

Token	Gold	Baseline	Hi-SEMFLOW
खुशी	B-SLANG	B-SLANG	B-SLANG
का	I-SLANG	I-SLANG	I-SLANG
ठिकाना	I-SLANG	O	I-SLANG
नहीं	I-SLANG	O	I-SLANG
रहा	I-SLANG	O	I-SLANG

Table 9: Recovery of full span boundary under flow refinement.

Here, the baseline prediction captures only the initial portion of the idiomatic phrase. Flow-based refinement enforces boundary continuity across all tokens, preventing premature termination. This illustrates the module's ability to maintain structural coherence even when intermediate tokens individually resemble non-slang usage.

Failure Mode: Mild Over-Extension भारत रत्न मिलना चाहिए (bhārat ratna milnā cāhiye; "should receive the Bharat Ratna")

Token	Gold	Baseline	Hi-SEMFLOW
भारत	B-SLANG	B-SLANG	B-SLANG
रत्न	I-SLANG	I-SLANG	I-SLANG
मिलना	O	O	I-SLANG
चाहिए	O	O	O

Table 10: Example of span over-extension under strong refinement.

In this case, structural smoothing slightly over-extends the predicted span. Such errors are comparatively rare and typically arise under strong curriculum schedules or large generator counts. Importantly, these errors preserve BIO validity and represent boundary calibration rather than structural violation.

Discussion Across examples, improvements primarily involve correction of internal span fragmentation and restoration of missing I-SLANG continuations. Failure cases involve limited over-extension rather than inconsistent transitions. These qualitative patterns support the quantitative findings: semantic flow functions as a soft structural regularizer, improving span-level coherence while maintaining competitive token-level discrimination.

8. Conclusion and Future Work

We introduced a Lie algebra–based semantic flow module for structured informal language identification in Hindi. Unlike discrete structured prediction approaches such as CRFs or post-hoc BIO decoding, our method models span consistency as a continuous geometric refinement of label distributions. Transition dynamics are parameterized via learnable antisymmetric generators and applied through a curriculum-controlled flow schedule, enabling structural regularization while remaining fully differentiable and architecture-agnostic.

Experiments on the `HiSlang-4.9k` benchmark show that semantic flow consistently reduces invalid BIO transitions and improves span-level F1, particularly for multi-token informal expressions. While macro-F1 gains are modest and encoder-dependent, structural improvements are stable across Hindi-pretrained models. Ablation studies further demonstrate that generator size, curriculum strength, and interaction with decoding-time constraints critically influence performance,

underscoring the importance of calibrated structural refinement. These results suggest that continuous geometric transitions provide a principled alternative to discrete structured decoding in sequence labeling. Rather than replacing contextual encoders, semantic flow complements them by explicitly modeling local transition dynamics.

Several directions naturally follow from this work including extending semantic flow to larger and multi-class label spaces, applying continuous structural refinement to other span-centric tasks such as chunking, event extraction, or argument mining would further test the generality of the geometric refinement framework.

Limitations

Although semantic flow improves structural consistency in span prediction, several limitations remain. First, the method introduces additional computational overhead due to matrix exponentiation. While the label space in this work is small ($|L| = 3$), scaling to larger label vocabularies may increase computational cost and memory usage. Efficient approximations or sparsity constraints may be necessary for high-dimensional label spaces. Second, performance gains are encoder-dependent. High-capacity multilingual models exhibit sensitivity to generator size and curriculum strength, and overly strong refinement may induce mild oversmoothing. This indicates that semantic flow requires careful hyperparameter calibration rather than functioning as a universally beneficial regularizer. Third, our evaluation is restricted to binary informal-span identification in Hindi. Although the formulation is architecture-agnostic, further validation is needed for multi-class NER settings, larger label inventories, and cross-lingual scenarios. Finally, improvements are more pronounced at the structural level (invalid transition reduction and span-level F1) than at token-level macro-F1. In applications where token-level accuracy is the sole objective, gains may appear modest despite improved span coherence. Further discussion on limited evaluation scope and statistical considerations is included in Appendix E.

Ethics Statement

This work builds upon the publicly available `HiSlang-4.9k` dataset, which contains manually annotated Hindi sentences collected from publicly accessible sources such as social media, subtitles, and online forums. No personally identifiable information was introduced as part of this study. Informal language and slang are inherently context-dependent and culturally nuanced. Certain expressions may carry social or cultural connotations that vary across communities. Our contribution focuses on modeling structural span consistency rather than interpreting, normalizing, or endorsing informal expressions. Any downstream deployment of models trained on informal language data should consider cultural sensitivity, domain appropriateness, and responsible usage. The semantic flow framework itself is task-agnostic and does not encode normative judgments about content.

Data and Code Availability

The data and code supporting this work are available upon request. Please contact `s23107@students.iitmandi.ac.in` for the same.

Acknowledgments

We sincerely thank the anonymous reviewers for their insightful comments and constructive suggestions. This work was conducted as part of an open collaboration between Thoughtworks and the BharatGen team at IIT Mandi.

References

SM Archana, Jay Prakash, Pramod Kumar Singh, and Waquar Ahmed. 2023. An effective biomedical named entity recognition by handling imbalanced data sets using deep learning and rule-based methods. *SN Computer Science*, 4(5):650.

David Belanger and Andrew McCallum. 2016. Structured prediction energy networks. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 983–992, New York, New York, USA. PMLR.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velivckovi'c. 2021. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *ArXiv*, abs/2104.13478.

Taco Cohen and Max Welling. 2016. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA. PMLR.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4115–4128.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 359–369.

Luca Falorsi, Pim de Haan, Tim R. Davidson, and Patrick Forré. 2019. Reparameterizing distributions on Lie groups. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pages 3244–3253. PMLR.

Jan E. Gerken, Jimmy Aronsson, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. 2023. Geometric deep learning and equivariant neural networks. *Artificial Intelligence Review*, 56(12):14605–14662.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Nikhil Gokul, Pushpak Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of EMNLP*.

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. Recent advances in named entity recognition: A comprehensive survey and comparative study.

Simran Khanuja, Diksha Bansal, Shashank Mehtani, Sarthak Khosla, Anshuman Dey, Naman Goyal, Parag Jain, Preksha Meena, Shubham Sharma, Karthik Sankaranarayanan, et al. 2021. Muril: Multilingual representations for indian languages. In *Findings of EMNLP*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Brian Lester, Daniel Pressel, Amy Hemmeter, Sagnik Ray Choudhury, and Srinivas Bangalore. 2020. Constrained decoding for computationally efficient named entity recognition taggers. *arXiv preprint arXiv:2010.04362*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Rudra Murthy, Pallab Bhattacharjee, Rahul Sharnagat, Jyotsana Khatri, Diptesh Kanojia, and Pushpak Bhattacharyya. 2022. Hiner: A large hindi named entity recognition dataset. *arXiv preprint arXiv:2204.13743*.

Badry Ali Mustofa and Wawan Laksito Yuly Saptomo. 2025. Use of natural language processing in social media text analysis. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, 4(2):1235–1238.

Zhengqi Pei, Zhewei Sun, and Yang Xu. 2019. Slang detection and identification. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pages 881–889.

Q-Success. 2024. Usage of hindi broken down by content management systems. Accessed: 2025-06-08.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third workshop on very large corpora*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009)*, pages 147–155.

Manikandan Ravikiran, Tanmay Tiwari, Vibhu Gupta, Rakesh Prakash, Rohit Saluja, and Shayan Mohanty. 2026. Do tokenizers fail on informal hindi expressions? evidence from static, downstream, and robustness analyses. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM)*, TBD. Association for Computational Linguistics. Workshop at EACL 2026.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the seventh named entities workshop*, pages 27–35.

Zhewei Sun, Qian Hu, Rahul Gupta, Richard Zemel, and Yang Xu. 2024. Toward informal language processing: Knowledge of slang

in large language models. *arXiv preprint arXiv:2404.02323*.

Simone Tedeschi, Federico Martelli, and Roberto Navigli. 2022. Id10m: Idiom identification in 10 languages. In *Findings of the Association for Computational linguistics: NAACL 2022*, pages 2715–2726.

Tanmay Tiwari, Vibhu Gupta, Manikandan Ravikiran, and Rohit Saluja. 2025. Hislang-4.9 k: A benchmark dataset for hindi slang detection and identification. In *Proceedings of the 8th International Conference on Natural Language and Speech Processing (ICNLSP-2025)*, pages 464–472.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 974–979.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. Conditional random fields as recurrent neural networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537.

A. Full `Hi-SEMFLOW` Architecture Overview

To facilitate understanding of the continuous structural refinement mechanism described in Section 3, we provide a complete architectural overview of the `Hi-SEMFLOW` pipeline in Figure 1. The diagram maps directly onto Sections 3.1–3.4 and illustrates how geometric transition operators interact with token-level predictions. Beyond procedural description, this appendix provides additional theoretical intuition for the geometric formulation underlying `Hi-SEMFLOW`.

A.1. Base Encoder and Independent Logits (§3.1)

Given an input sequence $X = (x_1, \ldots, x_T)$, a pretrained transformer encoder produces contextual representations

$$h_t = \text{Encoder}(X)_t.$$

A linear classifier generates independent token-level logits

$$z_t = Wh_t + b \in \mathbb{R}^{|\mathcal{L}|},$$

where $\mathcal{L} = \{\text{B-INF}, \text{I-INF}, \text{O}\}$.

These logits lie in a low-dimensional label space $\mathbb{R}^{|\mathcal{L}|}$. Before refinement, each z_t represents a locally computed distribution over BIO labels without explicit structural interaction. In contrast to CRFs, which impose discrete transition scores during decoding, we operate directly in logit space and treat label predictions as continuous vectors subject to smooth transformations.

This corresponds to the left branch of Figure 1 (Input $\rightarrow$ Encoder $\rightarrow$ Linear Classifier).

A.2. Context-Dependent Lie Algebra Generators (§3.1)

Structural refinement begins by constructing position-specific transition operators. A projection head predicts generator coefficients:

$$\alpha_t = f_\theta(h_t) \in \mathbb{R}^K.$$

We maintain K learnable generator matrices $G_k \in \mathbb{R}^{|\mathcal{L}|\times|\mathcal{L}|}$, which are antisymmetrized:

$$\widetilde{G}_k = \tfrac{1}{2}(G_k - G_k^\top).$$

Antisymmetric matrices form the Lie algebra of the orthogonal group. This constraint ensures that the infinitesimal transformations they generate are locally norm-preserving. Intuitively, this prevents structural refinement from arbitrarily scaling or distorting label confidences, instead encouraging structured rotations within label space.

The generators are combined into a position-specific infinitesimal operator:

$$A_t = \sum_{k=1}^{K} \alpha_{t,k} \widetilde{G}_k.$$

This construction corresponds to the central red branch of Figure 1 (Projection Head $\rightarrow$ Generators $\rightarrow$ Antisymmetrization $\rightarrow$ Weighted Combination).

A.3. Exponential Transition Operators (§3.1)

Finite transition operators are obtained via the matrix exponential:

$$F_t = \exp(A_t).$$

From a dynamical systems perspective, A_t defines an infinitesimal generator of motion in label space, and $\exp(A_t)$ integrates this motion into a finite transformation. This can be interpreted as

performing one step of a continuous-time flow on the label manifold.

The exponential ensures smooth, well-conditioned transformations and guarantees approximate orthogonality when generators are antisymmetric. In practice, we implement $\exp(A_t)$ using a truncated third-order Taylor approximation, which empirically provides a stable and efficient trade-off between accuracy and computational cost.

This corresponds to the "Matrix Exponential" block in the diagram.

A.4. Logit-Space Structural Propagation (§3.1)

Structural information is propagated left-to-right in logit space:

$$\widetilde{z}_t = F_t z_{t-1}.$$

This operation can be interpreted as transporting structural evidence along the sequence. Rather than assigning discrete transition scores, we transform the previous token's logit vector through a learned geometric operator.

The refined logits interpolate between local evidence and propagated structure:

$$\widehat{z}_t = (1 - \lambda_t) z_t + \lambda_t \widetilde{z}_t.$$

This interpolation acts as a convex combination between lexical evidence and structural coherence. For small λ_t, predictions are primarily local; as λ_t increases, structural consistency increasingly shapes span boundaries.

Final predictions are:

$$\widehat{p}_t = \text{softmax}(\widehat{z}_t).$$

These operations correspond to the green blocks in Figure 1 (Logit Propagation $\rightarrow$ Interpolation $\rightarrow$ Softmax).

A.5. Curriculum-Controlled Refinement (§3.2)

Refinement strength is gradually increased across epochs:

$$\lambda^{(e)} = \lambda_{\text{min}} + \frac{e}{E}(\lambda_{\text{max}} - \lambda_{\text{min}}).$$

From an optimization perspective, this curriculum stabilizes training by allowing the encoder to first learn lexical discrimination before introducing strong structural coupling. Gradual integration reduces the risk of early over-smoothing and helps avoid degenerate solutions where structural propagation dominates token-level evidence.

This schedule is depicted as the dashed curriculum branch controlling interpolation in Figure 1.

A.6. Geometric Regularization (§3.2)

We introduce magnitude, orthogonality, and temporal smoothness penalties:

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{mag}} + \beta \mathcal{L}_{\text{ortho}} + \gamma \mathcal{L}_{\text{temp}}.$$

These constraints serve complementary roles:

- Magnitude regularization limits excessive structural strength.
- Temporal smoothness encourages coherent transition dynamics across adjacent tokens.
- Orthogonality preservation compensates for approximation error in the truncated exponential and maintains well-conditioned operators.

Together, these penalties encourage stable geometric flows without introducing discrete decoding steps. These regularizers correspond to the "Flow Regularization" block in the lower-right portion of the diagram.

A.7. Optional BIO Constraints (§3.3)

For comparison with discrete structured decoding, decoding-time BIO masking may optionally be applied:

$$z_t^{\text{BIO}} = z_t + \log(\text{softmax}(z_{t-1})M).$$

This mechanism enforces legality through combinatorial constraints and is shown as a conditional branch in Figure 1. Importantly, `Hi-SEMFLOW` does not require discrete masking for structural coherence, but the comparison isolates the contribution of continuous geometric refinement.

A.8. Final Training Objective (§3.4)

The total training objective combines cross-entropy with geometric regularization:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \eta \mathcal{L}_{\text{flow}}.$$

From a broader perspective, `Hi-SEMFLOW` replaces discrete transition scoring with a continuous flow over label space, modeling BIO dependencies as smooth transformations rather than combinatorial constraints. Figure 1 summarizes this pipeline.

B. Additional Theoretical and Practical Considerations

This appendix addresses additional theoretical and practical aspects of `Hi-SEMFLOW`, including stability properties, gradient behavior, computational complexity, directional refinement, and scalability to larger label sets.

B.1. Stability of Logit-Space Refinement

We first analyze the stability of the refinement step:

$$\widehat{z}_t = (1-\lambda)z_t + \lambda F_t z_{t-1}.$$

Proposition 1 (Norm Stability). If A_t is antisymmetric ($A_t^\top = -A_t$), then $F_t = \exp(A_t)$ is orthogonal. Consequently,

$$\|F_t z_{t-1}\|_2 = \|z_{t-1}\|_2.$$

Proof Sketch. For antisymmetric A_t, we have $\exp(A_t)^\top = \exp(-A_t)$, implying

$$F_t^\top F_t = \exp(-A_t)\exp(A_t) = I.$$

Hence F_t preserves the ℓ_2 norm.

Corollary (Bounded Refinement). For $\lambda \in [0,1]$,

$$\|\widehat{z}_t\|_2 \le (1-\lambda)\|z_t\|_2 + \lambda\|z_{t-1}\|_2.$$

Thus refinement does not amplify logit magnitude beyond convex interpolation between adjacent tokens. This provides a theoretical justification for the empirical stability observed in training.

B.2. Gradient Behavior

Because F_t is approximately orthogonal, its spectral norm is close to 1. Therefore, the gradient of the propagated term

$$\tilde{z}_t = F_t z_{t-1}$$

does not introduce exponential amplification or vanishing across positions.

Furthermore, since refinement uses base logits z_{t-1} rather than recursively refined logits $\widehat{z}_{t-1}$, structural propagation does not accumulate multiplicatively along the sequence. This design choice prevents drift and ensures stable gradient flow.

B.3. Computational Complexity and Runtime Comparison

The additional computational cost arises from computing

$$F_t = \exp(A_t),$$

where $A_t \in \mathbb{R}^{|\mathcal{L}|\times|\mathcal{L}|}$.

Using a truncated third-order Taylor expansion, the per-token cost is

$$O(|\mathcal{L}|^3).$$

For our setting ($|\mathcal{L}| = 3$), this cost is negligible relative to encoder computation.

For comparison:

- Linear-chain CRF decoding via Viterbi has complexity $O(T|\mathcal{L}|^2)$.
- `Hi-SEMFLOW` refinement has complexity $O(T|\mathcal{L}|^3)$.

Since $|\mathcal{L}|$ is small in span-based tagging tasks, both approaches incur negligible overhead relative to transformer encoding. Empirically, we observe no measurable training slowdown compared to standard token classification.

B.4. Directional vs Bidirectional Refinement

We apply refinement left-to-right to align with the directional nature of BIO dependencies (e.g., `B-INF` typically precedes `I-INF`).

A bidirectional variant could refine using both z_{t-1} and z_{t+1}:

$$\tilde{z}_t = F_t^{(L)} z_{t-1} + F_t^{(R)} z_{t+1}.$$

While potentially beneficial, this introduces additional operators and increases computational cost. We leave systematic evaluation of bidirectional refinement to future work.

B.5. Extension to Larger Label Sets

The framework naturally generalizes to multi-class or multi-span tagging tasks.

For $|\mathcal{L}| > 3$:

- Generator matrices scale as $|\mathcal{L}| \times |\mathcal{L}|$.
- Antisymmetric parameterization still guarantees orthogonality.
- Matrix exponential remains well-defined.

Although computational cost scales as $O(|\mathcal{L}|^3)$, structured prediction tasks typically involve modest label sizes (e.g., BIO tagging, chunking, NER categories). Efficient approximations (e.g., low-rank generators) may further reduce cost in high-cardinality settings.

Thus, `Hi-SEMFLOW` is not restricted to binary slang identification and can extend to general span-centric structured prediction tasks.

C. Evaluation Metrics

We formally define the evaluation metrics used throughout the paper for clarity and reproducibility.

C.1. Token-Level Metrics

Let $\hat{y}_t$ denote the predicted label at position t, and y_t the corresponding gold label.

Model	Setting	Macro-F1	Span-F1	Invalid BIO (%)
MuRIL	Full Flow	0.9414	0.918	1.2
	$-\mathcal{L}_{\text{mag}}$	0.9378	0.912	1.9
	$-\mathcal{L}_{\text{ortho}}$	0.9396	0.915	1.6
	$-\mathcal{L}_{\text{temp}}$	0.9402	0.916	1.5
IndicBERTv2	Full Flow	0.9373	0.936	1.3
	$-\mathcal{L}_{\text{mag}}$	0.9339	0.930	2.0
	$-\mathcal{L}_{\text{ortho}}$	0.9354	0.933	1.7
	$-\mathcal{L}_{\text{temp}}$	0.9361	0.934	1.6
mBERT	Full Flow	0.9205	0.912	1.5
	$-\mathcal{L}_{\text{mag}}$	0.9168	0.906	2.2
	$-\mathcal{L}_{\text{ortho}}$	0.9189	0.909	1.9
	$-\mathcal{L}_{\text{temp}}$	0.9196	0.910	1.8

Table 11: Ablation of geometric regularization components across the top three encoders. Full Flow values correspond exactly to Tables 1 and 2.

Precision, Recall, and F1. For each label $\ell \in \mathcal{L}$:

$$\text{Precision}_\ell = \frac{TP_\ell}{TP_\ell + FP_\ell}, \qquad \text{Recall}_\ell = \frac{TP_\ell}{TP_\ell + FN_\ell},$$

$$\text{F1}_\ell = \frac{2 \cdot \text{Precision}_\ell \cdot \text{Recall}_\ell}{\text{Precision}_\ell + \text{Recall}_\ell}.$$

Macro-F1. Macro-F1 averages F1 scores across all labels:

$$\text{Macro-F1} = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \text{F1}_\ell.$$

This metric assigns equal weight to each label and is appropriate for class-imbalanced settings.

C.2. Span-Level F1

A span is defined as a maximal contiguous sequence beginning with `B-INF` followed by zero or more `I-INF` tokens.

Let $\mathcal{S}^{\text{pred}}$ denote the set of predicted spans and $\mathcal{S}^{\text{gold}}$ the set of gold spans.

A predicted span is counted as correct if and only if its start and end indices exactly match a gold span.

Span-level precision and recall are:

$$\text{Precision}_{\text{span}} = \frac{|\mathcal{S}^{\text{pred}} \cap \mathcal{S}^{\text{gold}}|}{|\mathcal{S}^{\text{pred}}|},$$

$$\text{Recall}_{\text{span}} = \frac{|\mathcal{S}^{\text{pred}} \cap \mathcal{S}^{\text{gold}}|}{|\mathcal{S}^{\text{gold}}|}.$$

Span-level F1 is computed as the harmonic mean of these quantities.

C.3. Invalid BIO Transition Rate

We measure structural validity using the proportion of illegal BIO transitions.

Let $(\hat{y}_{t-1}, \hat{y}_t)$ denote consecutive predicted labels. A transition is considered invalid if it violates BIO constraints (e.g., `I-INF` without a preceding `B-INF`).

The invalid BIO rate is defined as:

$$\text{Invalid BIO (\%)} = \frac{\sum_{t=2}^{T} \mathbf{1}\left[\text{transition invalid}\right]}{T-1} \times 100.$$

Lower values indicate stronger structural coherence.

C.4. Boundary Error Analysis

We categorize boundary errors into:

- **Fragmentation:** A gold multi-token span predicted as multiple shorter spans.
- **Over-extension:** A predicted span extends beyond the gold boundary.
- **Missed span:** A gold span predicted entirely as `O`.

These categories are computed by comparing predicted and gold span boundaries.

D. Ablation of Geometric Regularization Across Encoders

We analyze the contribution of individual geometric regularization components in

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{mag}} + \beta \mathcal{L}_{\text{ortho}} + \gamma \mathcal{L}_{\text{temp}}.$$

Experiments are conducted on the three strongest-performing encoders: MuRIL, IndicBERTv2, and mBERT. Each regularizer is

removed independently while keeping all other components fixed. (See Table 11)

Across all three encoders, removing magnitude regularization leads to the largest increase in invalid BIO transitions and the most consistent drop in span-level F1, indicating that controlling generator strength is critical for stable refinement. Orthogonality preservation contributes to structural validity under truncated exponential approximation, while temporal smoothness provides smaller but consistent improvements. The combined regularization yields the strongest structural coherence across models.

E. Additional Analysis

E.1. Quantifying Boundary Over-Extension

While semantic flow improves span continuity, it may occasionally introduce mild span over-extension, as illustrated in Table 10. To quantify the impact of such errors, we analyze trends in structural validity and span-level precision.

Across all evaluated encoders, Hi-SEMFLOW reduces invalid BIO transitions by up to **50%** (Table 2), while consistently improving span-level F1. Importantly, we do not observe any corresponding degradation in precision that would indicate systematic over-extension. This suggests that over-extension errors are relatively infrequent and localized.

Further, qualitative inspection indicates that such cases primarily arise under strong refinement settings (Section 7.3), where structural influence dominates local lexical evidence. These errors typically involve extension by a single token and preserve BIO validity, indicating boundary calibration rather than structural inconsistency.

E.2. Transition Error Rate vs. Hard Constraints

Unlike CRF-based models that enforce strict BIO legality through constrained decoding, Hi-SEMFLOW adopts a soft structural approach.

We quantify structural consistency using the *invalid BIO transition rate*. As shown in Table 2, Hi-SEMFLOW reduces invalid transitions substantially across models (e.g., IndicBERT: $3.8\% \rightarrow 1.9\%$, MuRIL: $2.6\% \rightarrow 1.2\%$).

This demonstrates that while Hi-SEMFLOW does not strictly forbid invalid transitions, it effectively *discourages* them through learned geometric transformations. This trade-off enables flexibility in ambiguous cases while maintaining strong structural coherence.

E.3. Sensitivity to Regularization Parameters

We analyze the role of flow regularization parameters through component ablations (Appendix D, Table 11).

Removing magnitude regularization (L_{mag}) results in the largest performance drop, indicating that controlling structural strength is critical for stable refinement. Similarly, removing orthogonality (L_{ortho}) or temporal smoothness (L_{temp}) leads to consistent degradation in both span-F1 and BIO validity.

These findings suggest that performance is sensitive to the balance of regularization terms, even without explicit grid search over β and γ. A full sensitivity analysis over parameter ranges is left for future work.

E.4. Analysis of Over-Smoothing Behavior

A potential limitation of continuous structural refinement is over-smoothing, where adjacent tokens may be incorrectly merged into a single span.

Empirically, such cases are rare and occur primarily under strong curriculum schedules or large generator counts (Section 6). Importantly, these errors differ from baseline failures: instead of producing fragmented or invalid spans, the model produces structurally valid but slightly over-extended spans.

This behavior suggests that semantic flow acts as a *soft structural regularizer*, prioritizing coherence over strict boundary precision in uncertain contexts.

E.5. Explanation for XLM-R Performance Trends

For XLM-R, we observe a slight decrease in macro-F1 (Table 1) despite improvements in structural metrics (Table 2).

This discrepancy arises because macro-F1 is sensitive to token-level label distribution, particularly the majority O class. Semantic flow primarily improves span boundary consistency, which benefits span-F1 and reduces invalid transitions, but may introduce minor shifts in token-level predictions.

This highlights a trade-off: structural refinement improves sequence-level coherence even when token-level accuracy remains unchanged or slightly decreases.

E.6. On Scope of Evaluation

Hi-SEMFLOW is evaluated on the HiSlang-4.9k dataset, which focuses on multi-token informal ex-

pressions in Hindi. This choice is intentional: the dataset emphasizes span consistency, making it suitable for evaluating structural refinement methods. While the framework is architecture-agnostic and applicable to broader sequence labeling tasks, extending evaluation to larger benchmarks (e.g., Hindi NER datasets with larger label spaces) is left for future work. We note that scaling to larger label sets increases computational cost due to matrix exponentiation ($\mathcal{O}(|L|^3)$), though efficient approximations (e.g., low-rank generators) can mitigate this.

E.7. On Statistical Significance and Multi-Seed Variance

All reported results correspond to single training runs for computational efficiency. While trends are consistent across models and metrics (e.g., reduction in invalid BIO transitions and improvements in span-F1), we acknowledge that multi-seed evaluation would provide stronger statistical confidence. We leave reporting of variance and confidence intervals as future work.

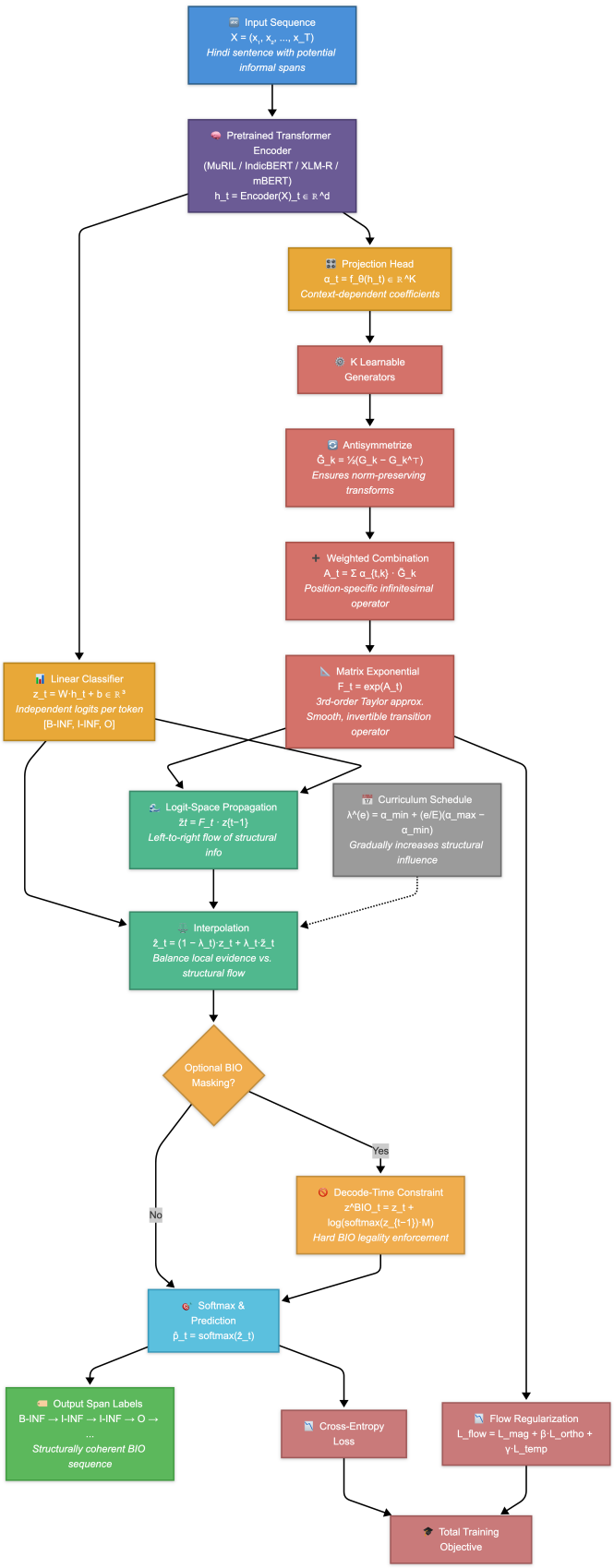


Figure 1: Complete `Hi-SEMFLOW` pipeline illustrating (i) independent token logits, (ii) context-dependent Lie algebra generator construction, (iii) matrix exponential transition operators, (iv) logit-space structural propagation with curriculum-controlled interpolation, (v) optional BIO masking, and (vi) joint cross-entropy and geometric regularization.

NeCCo: Nepali Cultural Commonsense Benchmark for Large Language Model Evaluation

Sanket Shrestha*, **Raunak Regmi,*** **Sadikshya Ghimire,***
Satyam Rana, Supriya Khadka
Sunway College Kathmandu, Birmingham City University, Kathmandu, Nepal
{sanket_24128469, raunak_regmi_a25, sadikshya_23140745}@sunway.edu.np
{satyam_rana_a25, supriya}@sunway.edu.np

Abstract

Large language models perform strongly on standard evaluations, yet these benchmarks prioritize high-resource languages and culturally dominant knowledge, leaving culture-specific commonsense underexamined. In low-resource languages such as Nepali, everyday communication depends on culturally embedded cues, including kinship hierarchies, ritual practices, food systems, idioms, and honorific distinctions that literal translation often fails to capture. As a result, models that appear competent on global metrics can perform poorly in local contexts. To address this gap, we introduce **NeCCo**, a curated multiple-choice benchmark for culturally situated reasoning across five domains: kinship and social hierarchy; festivals, rituals, and geography; idioms, proverbs, and metaphors; commonsense and daily life; and gastronomy, agriculture, and nature. The dataset was created through structured authoring, cross-review, and normalization, and is released in Devanagari, English, and Romanized formats. We evaluate multiple state-of-the-art LLMs using standardized prompting and controlled decoding. Results show substantial variation: models perform better on globally documented knowledge such as geography, but struggle with relational and linguistically implicit tasks, including extended kinship reasoning and proverb interpretation. The most culturally dense categories expose brittleness and increased hallucination. These findings suggest that multilingual competence requires more than translation coverage and highlight the need for culturally grounded benchmarks and training signals.

Keywords: Cultural Common sense, Low-Resource Languages, Multilingual Evaluation, Benchmark Construction, LLM Evaluation, Cross-Cultural Reasoning.

1. Introduction

Nepali is an Indo-Aryan language spoken by over 17 million people and serves as the official language of Nepal. Despite its widespread use, it is considered a low-resource language in NLP due to its limited presence in large-scale training corpora and evaluation benchmarks (Shahi and Sitaula, 2022). This under-representation leads to insufficient modeling of linguistic variability and culturally grounded nuances in large language models (LLMs) (Nyachhyon et al., 2025). Many multilingual benchmarks either focus on high-resource languages or test translation-equivalent skills, which can obscure whether models actually understand locally grounded meanings in languages such as Nepali (Hu et al., 2020; Liang et al., 2020; Nyachhyon et al., 2025).

A key source of difficulty is cultural common sense: knowledge that speakers routinely use but rarely state explicitly. In Nepali, interpretation often depends on kinship structure and social hierarchy, honorific choice, ritual etiquette, and idiomatic language that does not survive literal translation (Krause, 1980). For example, kinship terms encode distinctions such as paternal versus maternal relations and elder versus younger relatives, and these distinctions shape appropriate address forms and expected behavior. Such cues are sparsely documented in globally available text and are therefore weakly represented in the data that dominates LLM training. As a result, a model can appear strong on general multilingual benchmarks while still failing on culturally specific reasoning required in everyday Nepali use (Thakur, 2025).

This paper targets that evaluation blind spot. We introduce NeCCo, a multiple-choice benchmark designed to test culturally anchored commonsense reasoning in Nepali across five domains: kinship and social hierarchy; festivals, rituals, and geography; idioms, proverbs, and metaphors; commonsense and daily life; and gastronomy, agriculture, and nature. We adopt a multiple-choice format to support controlled comparisons across models and prompt settings, reduce sensitivity to open-ended generation style, and enable fine-grained analysis of systematic confusions. At the same time, the format keeps the evaluation focused on selecting culturally appropriate inferences rather than on fluency or long-form explanation quality.

NeCCo is authored by native speakers and cross-reviewed to reduce ambiguity, with standardized formatting to support consistent evaluation. Using this benchmark, we evaluate a set of state-

*Equal contribution

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 154–168
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

of-the-art LLMs under standardized prompting and decoding, and analyze performance by category to identify where culturally grounded reasoning breaks down (Thapa et al., 2025). We then use these findings to highlight gaps in current multilingual evaluation practice and to motivate more culturally grounded training and assessment.

Our core contributions are

- We introduce **NeCCo**, a culturally anchored multiple-choice benchmark for evaluating Nepali commonsense reasoning across five domains.
- We conduct a systematic evaluation of contemporary LLMs on culturally embedded reasoning tasks, highlighting failures that standard multilingual benchmarks can miss.
- We provide recommendations for future dataset construction and model training that incorporate culturally grounded signals.

2. Related Work

2.1. Multilingual and Cross-Lingual Evaluation of LLMs

A significant portion of LLM evaluation continues to rely on English-centric benchmarks. In these settings, multilingual capability is frequently inferred through translation or limited cross-lingual transfer rather than direct evaluation in the target languages (Clark et al., 2020). Broad multilingual benchmarks, such as XTREME-style evaluation suites and information-seeking datasets like TyDi QA (Clark et al., 2020), were introduced to assess generalization across typologically diverse languages. These evaluations consistently show significant performance degradation outside high-resource settings.

More recent evaluation frameworks, including holistic taxonomies and large collaborative benchmarks like BIG-bench (Kazemi et al., 2025), emphasize the multi-dimensional nature of model quality. In these frameworks, aggregate accuracy can easily obscure failures in robustness, calibration, fairness, and cultural coverage. Multilingual extensions of complex reasoning benchmarks demonstrate that performance drops persist even under few-shot prompting. This indicates that multilinguality is not a uniform capability. Instead, it is a fragile interaction between training data distribution, script variation, and socio-cultural grounding (Clark et al., 2020).

2.2. Commonsense Reasoning Benchmarks

Traditionally, commonsense reasoning benchmarks have focused on general-purpose inference, encompassing causal, temporal, physical, and social dimensions. However, widely used datasets such as CommonsenseQA, SocialIQA, and PIQA (Sap et al., 2019) are primarily constructed in English and implicitly encode Western or globally dominant knowledge priors (Toroghi et al., 2025). Cross-lingual efforts like XCOPA (Ponti et al., 2020) reveal that transferring commonsense tasks across languages exposes distinct linguistic and cultural limitations. In many cases, translation-based approaches actually outperform direct multilingual reasoning (Parashar et al., 2025).

This highlights a major limitation: most existing benchmarks evaluate generalized commonsense rather than culturally situated reasoning. Key dimensions like kinship structures, honorific systems, ritual practices, and idiomatic expressions remain starkly underrepresented. This creates a critical evaluation gap where models appear competent in abstract reasoning but fail in culturally grounded contexts.

2.3. Cultural and Socio-Cultural Bias in LLMs

A parallel body of work investigates cultural and socio-cultural bias in LLMs, demonstrating that models inherently encode and reproduce patterns present in their training data. Foundational benchmarks such as CrowS-Pairs (Nangia et al., 2020) and StereoSet (Nadeem et al., 2021) measure stereotypical associations, BBQ (Parrish et al., 2022) evaluates bias in question answering, and HolisticBias (Zake, 2023) expands this coverage across multiple demographic dimensions. Although essential for assessing fairness, these tools are largely centered on English-speaking or Western contexts. This geographical and linguistic concentration leaves significant gaps in understanding localized harms, such as the specific mechanisms of gender bias in under-represented languages like Nepali (Khadka and Bhattarai, 2025). Emerging work on culturally diverse evaluation highlights that true cultural competence extends beyond simple bias mitigation. It requires a nuanced understanding of implicit norms, localized knowledge systems, and culturally bounded meanings (Piao et al.). When confronted with unfamiliar cultural inputs in these settings, LLMs frequently exhibit hallucination, over-generalization, or default to globally dominant interpretations (Peters et al., 2022).

2.4. Low-Resource and Nepali NLP

Low-resource languages present a unique set of challenges characterized by limited high-quality corpora, sparse benchmark availability, and severe under-representation in large-scale training pipelines. Nepali exemplifies this pattern: despite being spoken by millions, it remains critically under-resourced in terms of standardized evaluation frameworks (Cahyawijaya, 2024). Existing efforts in Nepali NLP have primarily focused on foundational tasks, including part-of-speech tagging (Pradhan and Yajnik, 2024), named entity recognition (Singh et al., 2019), machine translation (Shahi and Sitaula, 2022), text classification (Prabha et al., 2018) and text-to-speech (Khadka et al., 2023). Benchmarks such as Nep-gLUE (Timilsina et al., 2022) and subsequent Nepali NLU datasets (Nyachhyon et al., 2025) have contributed to task-level standardization. Concurrently, monolingual models like NepBERTa have demonstrated improved performance over multilingual baselines in specific settings (Timilsina et al., 2022). Additionally, resources such as FLoRes provide reliable evaluation datasets for Nepali in machine translation (Guzmán et al., 2019). However, these efforts largely target conventional syntactic or semantic tasks. Much of Nepali reasoning is transmitted through oral tradition and lived experience rather than formal corpora, making it particularly difficult for models to learn through these standard, foundational training datasets.

2.5. Toward Culturally Grounded Evaluation

Recent work has increasingly highlighted the limitations of large language models in capturing culturally grounded reasoning, particularly beyond high-resource and globally dominant contexts. While several benchmarks have been proposed for South Asian languages, most focus on broader Indic settings or emphasize translation-based evaluation rather than culturally embedded understanding. Datasets such as SANSKRITI (Maji et al., 2025) and MILU (Verma et al., 2025) evaluate LLMs across multiple Indic languages and domains, demonstrating that model performance drops in culturally rich categories such as rituals, social norms, and traditional knowledge (Joshi et al., 2025). Similarly, large-scale multilingual evaluations like PARIKSHA reveal inconsistencies and reduced reliability of LLMs in low-resource languages, particularly when tasks require implicit reasoning or contextual awareness. However, these efforts largely aggregate across languages and do not account for the unique sociocultural structures of individual languages such as Nepali, as cultural common-sense in Nepali remains underrepresented in existing benchmarks. (Chiu et al., 2025).

3. Methodology

3.1. Dataset Creation

Dataset Overview The Nepali Cultural & Commonsense Benchmark (**NeCCo**) is a curated dataset of 1,295 four-option multiple-choice questions (A to D) designed to evaluate culturally grounded reasoning in Nepali.[1] Each question has a single unambiguous correct answer. The dataset emphasizes implicit cultural knowledge that cannot be inferred through direct translation or surface-level reasoning, and the distractors are written to be culturally plausible without requiring external tools. Each instance in the dataset follows a structured tabular format containing a unique ID, category, question prompt, four options, and the ground truth answer to facilitate systematic evaluation.

Linguistic Representations To study representation sensitivity and translation effects, each question is released in three aligned forms: native Devanagari Nepali, Romanized Nepali, and English translation. Romanized Nepali refers to the phonetic transcription of the native Devanagari script into the Latin alphabet, a format widely used in informal digital communication (Madhani et al., 2023). Unlike standard Devanagari, Romanized text often lacks rigid spelling conventions, introducing natural linguistic variability. For example, the Devanagari word राम्रो (good) is commonly romanized as *ramro*, and a phrase like के छ? (What's up?) might appear variously as *ke cha*, *k cha*, or *k chha*.

We provide English translations for categories where translation does not substantially distort the intended reasoning signal. Accordingly, the *Kinship & Social Hierarchy* and *Idioms, Proverbs & Metaphors* categories intentionally omit English translations to preserve semantic nuance. In total, NeCCo contains 3,460 released items: 1,295 (Devanagari) + 1,295 (Romanized) + 870 (English).[2]

Category Design and Distribution NeCCo is organized into five categories chosen to reflect core dimensions of Nepali life, social organization, ritual practices, and subsistence patterns. The

[1] NeCCo is publicly available at https://github.com/Mezamenta-Nakama09/NeCCo.

[2] The 870 English items correspond to the 1,295 total questions minus the 425 questions in the two non-translated categories.

dataset is approximately balanced, with each category contributing roughly 18 to 22% of the total questions (see Table 1).

Category	Questions
Kinship & Social Hierarchy	235
Festivals, Rituals & Geography	382
Idioms, Proverbs & Metaphors	190
Commonsense & Daily Life	200
Gastronomy, Agriculture & Nature	288

Table 1: Number of questions per category in NeCCo.

- **Kinship & Social Hierarchy:** Tests understanding of intricate familial relations, hierarchical distinctions, honorific usage, and lineage (maternal vs. paternal) reflecting Nepal's social structure.
- **Festivals, Rituals & Geography:** Covers religious observances (e.g., Dashain, Tihar), local jatras, agricultural calendars, ritual objects, and region-specific events shaped by Nepal's terrain.
- **Idioms, Proverbs & Metaphors:** Focuses on culturally embedded expressions whose meanings cannot be derived from literal translation, requiring contextual interpretation.
- **Commonsense & Daily Life:** Captures implicit norms in daily interactions, including etiquette, hospitality, purity concepts, and socially appropriate household customs.
- **Gastronomy, Agriculture & Nature:** Encompasses culinary practices, ingredient pairings, agricultural planting cycles, common tools, and culturally significant flora and fauna.

This categorization creates a natural gradient of difficulty. Categories such as Kinship and Idioms are inherently more challenging due to their reliance on relational reasoning and non-literal meaning, whereas Festivals and Geography include a mix of factual and contextual knowledge. Representative examples from each category are provided in Appendix A.

Data Collection and Curation Pipeline The dataset was collaboratively developed by four native Nepali speakers. Each contributor led specific categories, generating over 250 questions to ensure both domain specialization and diverse coverage. Questions were sourced from a combination of public references (e.g., Wikipedia, books, grammar materials) alongside rich oral knowledge from community elders and lived cultural experience, ensuring authenticity across regional and ethnic contexts.

To ensure dataset quality, maintain consistency, and minimize annotation bias, we adopted a multi-stage pipeline:

- **Initial Authoring:** Native speakers generated culturally grounded MCQs within their assigned domains.
- **Cross-Review:** Question sets were exchanged for peer review. Reviewers labeled each item as *Keep*, *Modify*, or *Delete* based on cultural accuracy, ambiguity, and distractor plausibility.
- **Consensus Revision:** Items marked *Modify* were iteratively refined, and disputed cases were resolved through discussion until consensus was reached.
- **Translation and Alignment:** Initial English translations were generated using machine translation tools (e.g., Google Translate[3]) and subsequently refined manually by bilingual native speakers to ensure semantic fidelity. Romanization was also standardized.
- **Post-processing and Quality Control:** We conducted manual inspections to correct spelling and grammatical issues. A custom automated cleaning script detected duplicate entries, missing values, CSV formatting inconsistencies, and sequencing errors.
- **Final Cross-Check:** A final review ensured alignment across the Devanagari, Romanized, and English variants.

3.2. Experimental Setup

3.2.1. LLMs Used

We evaluate nine recent LLMs chosen to span a range of architectures, release types (proprietary vs. open-weight), and multilingual capabilities. Model selection was informed by contemporaneous standings on OpenRouter and other public leaderboards, with an emphasis on general instruction following and multilingual reasoning performance at the time of evaluation.[4]

The evaluated models are: GPT-4o-mini, GPT-5.2, and GPT-OSS-120B (OpenAI) (Hesham and Hamdy, 2024; Ma et al., 2026; Agarwal et al., 2025); Gemini 3 Flash Preview (Google DeepMind) (Zhang et al., 2026); DeepSeek V3.2 (Liu et al., 2025); Grok 4.1 Fast (xAI) (Ma et al., 2026); GLM-5 (Tsinghua University and Zhipu AI) (Zeng et al., 2026); Minimax M2.5 (Ko, 2026); and Trinity Large Preview (OpenRouter, n.d.). We include

[3] https://translate.google.com/
[4] https://openrouter.ai/rankings

Prompt Format (2-shot MCQ)

System/Instruction: You are a helpful assistant that answers multiple choice questions about Nepali common sense and cultural etiquette. Your task is to provide ONLY the letter (A, B, C, or D) corresponding to the correct option. Do NOT provide any reasoning, explanation, or additional text. Just the single letter.

Example 1:
Question: तिमी आफन्तको घरमा खाना खाँदै छौ। काँक्रोको अचार एकदम पिरो छ। चलन अनुसार तिमीले के गर्छौ?
A: मुखबाट 'आफ्, आफ्' आवाज निकाल्दै पानी माग्छौ।
B: थोरै भात वा चिउरासँग मुछेर चुपचाप निल्छौ।
C: अचारलाई थालको छेउमा थुक्छौ।
D: पाहुनालाई गाली गर्छौ।
Answer: B

Example 2:
Question: तिमी मन्दिर भित्र छौ। देउतालाई ढोग्दा तिम्रो टाउकोले केमा छुने कोसिस गर्छौ?
A: मन्दिरको सिँढीमा।
B: देउताको पाउ वा तोकिएको चरण-पादुकामा।
C: घण्टीमा।
D: पुजारीको हातमा।
Answer: B

Evaluation Instance Template:
Question: {question}
A: {A}
B: {B}
C: {C}
D: {D}
Answer:

Figure 1: Prompt Template

Trinity Large Preview in particular because it accounts for the highest usage share for Nepali on OpenRouter (see Appendix C), making it a highly relevant, community-preferred baseline for culturally grounded Nepali evaluation.

3.2.2. Evaluation Procedure

We use a standardized few-shot prompting setup (2-shot) for all models. Two fixed example question-answer pairs are prepended to every evaluation instance across all categories. Keeping the demonstrations and instructions constant ensures that performance differences primarily reflect the model's reasoning, rather than variation in prompt framing. The prompt template and demonstrations are shown Figure 1.

All models are evaluated on the same four-way MCQ task and are instructed to output *only* the option letter. We use a uniform system prompt and identical input formatting for all models to minimize differences due to verbosity, output style, or prompt interpretation, and to focus the evaluation on answer selection.

Inputs are evaluated in three representations: Devanagari Nepali, Romanized Nepali, and English translation (except for the two categories where English is intentionally omitted). All models are accessed through a unified API interface. Decoding settings are kept consistent across models by using a single set of parameters, with no model-specific tuning. This standardized configuration ensures a fair evaluation, isolating performance differences to the models' inherent capabilities rather than external optimization. Each question is submitted exactly once per model and representation, with no resampling or response averaging.

3.2.3. Analysis Protocol

We evaluate model performance using **accuracy**, measured as the proportion of questions for which a model selects the correct option. Beyond overall accuracy, we report results **by category** to better capture domain-specific strengths and weaknesses, since aggregate scores can mask substantial variation across cultural domains.

We further stratify performance across the three input representations: **Devanagari Nepali**, **Romanized Nepali**, and **English translation**. This breakdown enables a direct assessment of how sensitive models are to changes in script and language form. For culturally dense categories such as *Kinship & Social Hierarchy* and *Idioms, Proverbs & Metaphors*, we do not include English-based analysis because English translations were intentionally omitted to avoid loss of meaning and evaluation ambiguity.

Finally, we compare performance across models, categories, and representations to identify recurring error patterns and question types that are systematically challenging. Although our primary metric is accuracy, these structured slices provide a clearer view of model limitations and highlight where cultural knowledge and language representation most strongly influence performance.

4. Results

In this section, we summarize overall trends in accuracy to highlight where current LLMs succeed and where they systematically fail on Nepali cultural commonsense. A complete breakdown of per-model results, including accuracy, precision, recall, and F1 across categories and representations, is provided in the Appendix B.

Table 2: Cross-Model Comparative Accuracy across Categories (Bold indicates highest in each category)

Model	Agriculture	Cultural Etiquette	Festivals	Geography	Kinship & Social Hierarchy	Idioms/Proverbs	Overall Avg
Gemini 3 Flash	0.447	**0.935**	0.821	0.570	**0.821**	**0.866**	**0.743**
GPT-5.2	0.448	0.930	**0.865**	**0.689**	0.694	0.539	0.694
GLM-5	**0.512**	0.886	0.740	0.517	0.706	0.779	0.690
Grok-4.1 Fast	0.508	0.879	0.742	0.520	0.702	0.768	0.687
GPT-4o Mini	0.419	0.824	0.592	0.504	0.534	0.621	0.582
MiniMax-M2.5	0.349	0.826	0.635	0.465	0.532	0.579	0.564
DeepSeek-V3.2	0.331	0.836	0.686	0.385	0.551	0.524	0.552
GPT-OSS-120B	0.357	0.774	0.581	0.544	0.517	0.539	0.552
Trinity large	0.297	0.866	0.572	0.270	0.455	0.421	0.480

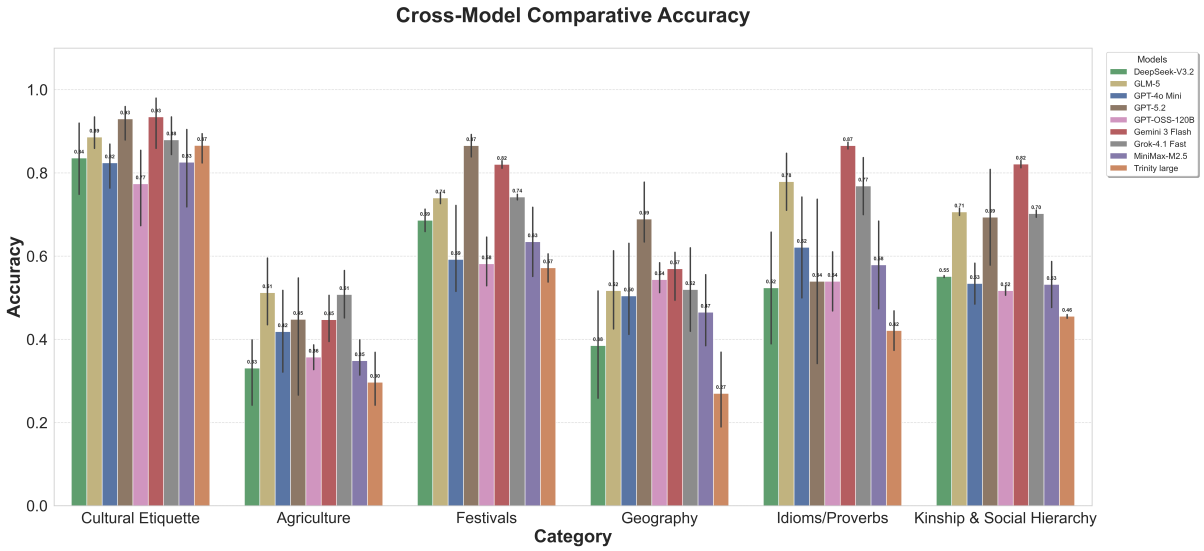


Figure 2: Comparative Accuracy across Categories

Comparative Accuracy across Categories. Figure 2 and Table 2 illustrate a significant imbalance in how models capture cultural knowledge across different domains. Models perform exceptionally well on widely documented domains. For instance, Cultural Etiquette is the best performing category, with both GPT-5.2 and Gemini 3 Flash achieving peak accuracies of 0.93. Festivals also demonstrate strong cross-model performance, with top models consistently scoring above 0.80.

Conversely, performance drops sharply on localized, community-specific domains like Agriculture and Geography. Agriculture is the most universally challenging category. Even the highest-performing models struggle in this area, with scores hovering between 0.30 and 0.51. This suggests a widespread deficit in implicit or rural localized knowledge. Furthermore, categories like Idioms/Proverbs and Kinship & Social Hierarchy are highly model-dependent. Gemini 3 Flash excels in Idioms/Proverbs (0.87) and Kinship (0.82), whereas models like Trinity large and DeepSeek-V3.2 struggle significantly in these exact same categories (scoring between 0.42 and 0.55). This variance indicates that specific training mixtures allow certain models to grasp complex linguistic and social nuances that others miss.

Performance Comparison: English vs. Nepali vs. Romanized. Figure 3 isolates the role of script and language form by evaluating each question across three formats. While aggregate trends favor native script, individual models exhibit fascinating divergence in their linguistic handling. Models such as Gemini 3 Flash and GLM-5 perform best in native Devanagari (0.83 and 0.78 respectively), outperforming their own English baselines. However, this is not a universal rule. GPT-5.2 demonstrates an inverted proficiency, performing best in English (0.78), followed by Romanized Nepali (0.75), and performing worst in native Devanagari (0.66). A similar trend is visible with GPT-4o Mini. This highlights that multilingual alignment varies heavily by model architecture, and language representation choices materially affect how reliably cultural commonsense is retrieved.

Overall Language Performance Across All Models. Figure 4a consolidates the linguistic results across all models to confirm a stable over-

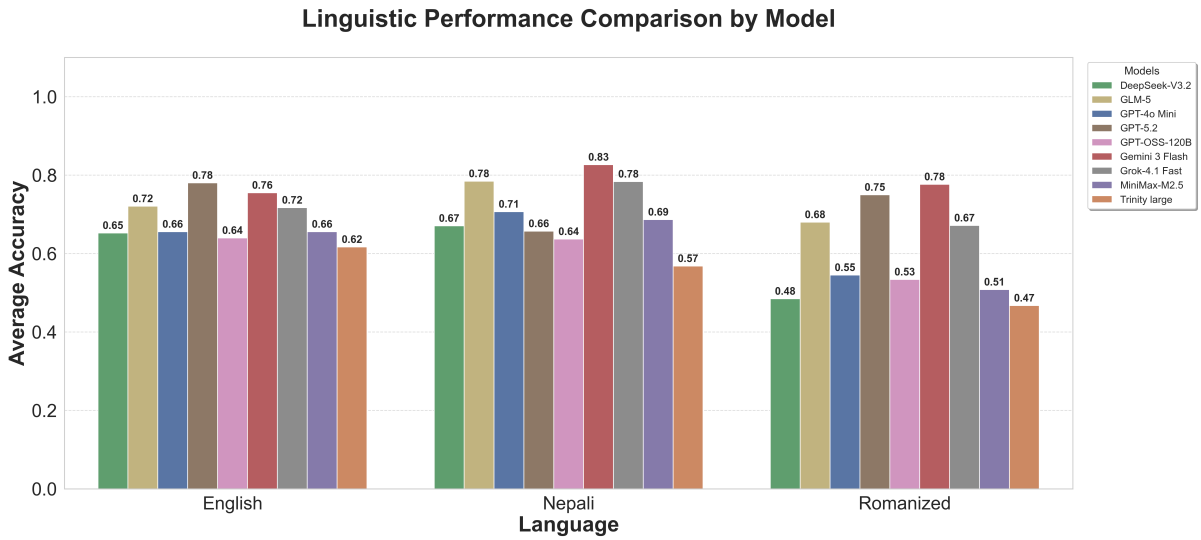
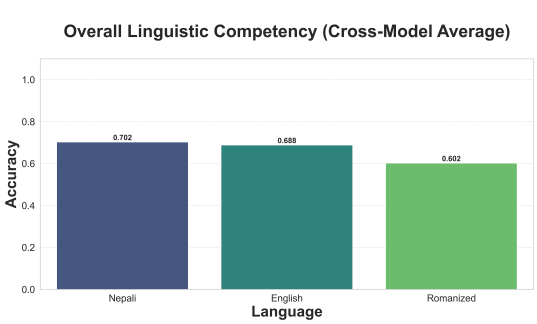
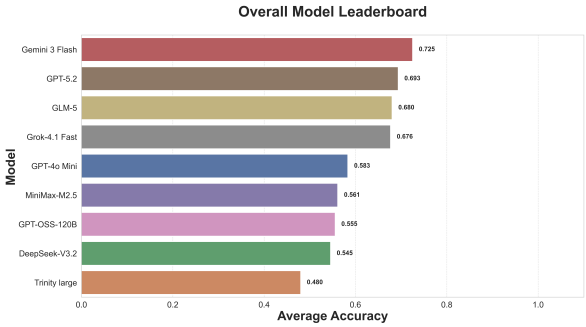


Figure 3: Performance Comparison: English vs. Nepali vs. Romanized



(a) Overall Language Performance Across All Models



(b) Overall Model Ranking (Mean Accuracy Across All Categories)

Figure 4: Aggregate results across input representations and models.

all ordering. Native Devanagari Nepali achieves the highest aggregate accuracy (0.702), followed closely by English (0.688). Romanized Nepali represents a significant step down (0.602). This aggregate pattern suggests that non-standard transliteration meaningfully hinders an LLM's access to its culturally grounded knowledge base.

Overall Model Ranking. Finally, Figure 4b presents the overall leaderboard, revealing a stratification in model capabilities. The top tier of models demonstrates a clear advantage, with Gemini 3 Flash leading the benchmark at an average accuracy of 0.725. This is followed closely by GPT-5.2 (0.693), GLM-5 (0.680), and Grok-4.1 Fast (0.676). There is a notable drop-off to the second tier of models, beginning with GPT-4o Mini (0.583) and cascading down to Trinity large (0.480). This distribution indicates that larger parameter counts or general popularity do not automatically guarantee superior cultural reasoning, as highly optimized models often outperform larger alternatives.

5. Discussion

The NeCCo evaluation offers a detailed view of how well current LLMs capture Nepali cultural intelligence. While the top-performing systems show strong general capabilities, we observe recurring failure modes that expose the limits of today's models in culturally grounded commonsense reasoning. In many cases, models handle surface-level facts and linguistic form reliably but struggle when correct answers depend on implicit norms, context-specific practices, or culturally encoded meanings that native speakers take for granted.

5.1. The Importance of Cultural Benchmarking

Most standard NLP benchmarks emphasize linguistic competence, general knowledge, or skills that transfer well across languages. They rarely test the everyday commonsense and social expectations that are obvious within a community but

opaque to outsiders. NeCCo illustrates why culturally targeted benchmarks are necessary: models can appear highly capable on global tasks while still failing in realistic local scenarios where appropriate behavior, social roles, and cultural interpretations matter.

This has practical implications for deployment. A model may produce fluent Nepali and handle translation accurately, yet remain unreliable when advising users in culturally sensitive contexts such as kinship address terms, social hierarchy, ritual practices, or community norms. Evaluating these dimensions helps prevent a form of cultural erasure in which systems are treated as "Nepali-capable" based primarily on language form, even though they lack key cultural competencies required for safe and useful real-world interaction.

Our results also underscore that cultural understanding does not automatically transfer through translation. The performance gap between English and Devanagari Nepali suggests that knowing a concept in English is not the same as recognizing its lived meaning, associated norms, and situational usage in Nepali contexts. Cultural competence therefore requires more than mapping words across languages. It requires interpreting meaning within local practices, expectations, and history.

5.2. Key Findings and Observations

Linguistic Representation and Script Sensitivity A consistent pattern in our findings is the clear advantage of native-script inputs. For several top-tier models, Devanagari Nepali performs as well as or better than English. This indicates that culturally grounded associations are often more accessible in the original language form, as native script provides stable token representations that map directly to culturally specific semantic spaces in the model's pre-training data.

In contrast, Romanized Nepali severely underperforms across models, showing a substantial drop in accuracy. A primary mechanical reason for this is high token fertility: standard LLM tokenizers often fail to recognize Romanized Indic words, breaking them down into excessively small, semantically disjointed character fragments. Furthermore, informal Romanization introduces significant orthographic noise and spelling variations in real-world use. This fragmentation and inconsistency make the model's conceptual retrieval highly brittle, weakening its ability to connect transliterated forms to stable cultural concepts.

We also observe a clear gap in figurative and culturally embedded language. Performance is notably low on Proverbs/Idioms and other idiomatic content, where meaning is rarely recoverable from literal interpretation. Correct answers often require familiarity with social context, implied intent, irony, and shared cultural references. These results suggest that current LLMs still struggle when the task requires inferring non-literal meaning that is culturally conventional rather than textually explicit.

Model Scaling and Local Knowledge Gaps Another important takeaway is that model scale alone does not guarantee better cultural reasoning. In our experiments, optimized proprietary models often outperform substantially larger open-weight models. This suggests that data quality, local coverage, and the presence of culturally representative training signals matter far more than raw parameter counts for low-resource cultural competence.

Finally, the weakest categories point to a severe local knowledge void. Low scores in areas such as agriculture, geography, and kinship relations indicate that many models lack grounded knowledge about Nepali seasonal routines, tools, local topography, and family relationship systems. These domains are often under-represented in widely used training corpora, which tend to over-index on globally dominant and frequently documented contexts. NeCCo makes these gaps visible and provides a concrete target for improving cultural coverage in future model training and evaluation.

5.3. Future Implications

NeCCo points toward a concrete direction for the next generation of Nepali-centered AI. Our findings suggest that improving cultural understanding is less about scaling model size and more about deliberately incorporating local knowledge and sustained feedback from native speakers throughout data curation, training, and evaluation. Future work should systematically investigate the effects of parameter tuning, prompt design, and targeted fine-tuning on Nepali cultural resources, including oral narratives, proverbs, and ethnographic texts. This domain-adaptive approach may help mitigate the severe performance gaps observed in categories like agriculture and kinship. Increasing parameter counts or relying strictly on translation pathways cannot compensate for missing culturally grounded signals.

Without such intentional effort, the digital divide risks expanding beyond access into a divide in representation and understanding. Communities may be able to use AI systems yet still be misunderstood when those systems interpret their language, traditions, and everyday practices through an incomplete cultural lens. Progress in AI for low-resource languages therefore requires inclusion: deeper cultural coverage, community-informed development, and evaluation methods that measure what truly matters in real use.

6. Conclusion

We introduced NeCCo, a culturally grounded benchmark for evaluating Nepali commonsense reasoning across domains shaped by social structure, language, and lived practice. Our evaluation demonstrates that strong performance on global benchmarks does not reliably translate to culturally grounded competence. Models consistently struggle with implicit and relational knowledge, particularly in domains like kinship, proverbs, and localized agriculture. Furthermore, results vary substantially based on linguistic representation. Native Devanagari generally outperforms English, while Romanized Nepali consistently degrades model performance. Overall, NeCCo highlights that advancing cultural reasoning depends heavily on high-quality, culturally representative data and targeted evaluation rather than simply increasing model scale. This benchmark provides a critical foundation for more inclusive and accurate assessments of LLMs in low-resource settings.

7. Limitations

While NeCCo introduces a novel framework for cultural evaluation, it has several limitations. First, the dataset is relatively small; despite careful curation, a larger corpus would support more granular statistical analysis and better capture Nepal's diverse subcultures. Second, the use of fixed multiple-choice questions enables standardized evaluation but constrains the assessment of open-ended cultural reasoning and may oversimplify complex lived knowledge. The benchmark also focuses on a limited set of foundational domains. Future work should expand into areas such as indigenous folklore, regional histories, and contemporary digital culture for a more comprehensive evaluation. Additionally, annotations were conducted solely by the authors, who are native Nepali speakers. While this ensures linguistic fluency, it introduces potential subjectivity and lacks input from domain experts such as sociologists or anthropologists. The absence of inter-annotator agreement metrics (e.g., Cohen's Kappa or Krippendorff's Alpha) further limits the ability to assess annotation reliability.

Furthermore, our evaluation relies on standardized default inference settings and a fixed 2-shot prompt. Model performance on culturally grounded reasoning can be sensitive to temperature adjustments or advanced prompt engineering, such as providing dynamic, domain-specific cultural demonstrations. Finally, the benchmark does not randomize answer option order across runs, leaving room for positional bias in model responses. Prior work suggests that LLMs may favor certain option positions independent of content, introducing evaluation noise (Pezeshkpour and Hruschka, 2024). Future iterations will incorporate option randomization and repeated evaluations to better reflect true reasoning performance.

8. Ethical Consideration

NeCCo is constructed using culturally grounded knowledge sourced from native Nepali speakers, which inherently introduces challenges related to representation, bias, and coverage. While we employed a multi-stage validation process to ensure clarity and cultural accuracy, the dataset may still reflect localized perspectives and might not fully capture the vast diversity of Nepal's socio-cultural landscape. Additionally, reducing culturally specific practices to fixed-answer formats risks flattening context-dependent knowledge. This benchmark is intended strictly for evaluation and research purposes; it should not be interpreted as a complete, definitive, or monolithic representation of Nepali culture. We strongly encourage responsible use, transparency, and future expansion through broader community involvement to continuously improve the benchmark's inclusivity and cultural fidelity.

9. Data/Code Availability Statement

All the data and code supporting the findings of this study is publicly available on GitHub.

10. References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Xiaoxuan Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott

McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, et al. 2025. gpt-oss-120b and gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Samuel Cahyawijaya. 2024. *Llm for everyone: Representing the underrepresented in large language models*. Hong Kong University of Science and Technology (Hong Kong).

Yu Ying Chiu, Liwei Jiang, Bill Yuchen Lin, Chan Young Park, Shuyue Stella Li, Sahithya Ravi, Mehar Bhatia, Maria Antoniak, Yulia Tsvetkov, Vered Shwartz, and Yejin Choi. 2025. Culturalbench: A robust, diverse, and challenging cultural benchmark by human-ai culturalteaming.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali–english and sinhala–english. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6098–6111.

Alaa Hesham and Abeer Hamdy. 2024. Fine-tuning GPT-4o-Mini for programming questions generation. In *Proceedings of the 2024 International Conference on Computer and Applications (ICCA)*, pages 1–6.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International conference on machine learning*, pages 4411–4421. PMLR.

Neha Joshi, Pamir Gogoi, AasimBaig Mirza, Aayush Jansari, Aditya Yadavalli, Ayushi Pandey, Arunima Shukla, Deepthi Sudharsan, Kalika Bali, and Vivek Seshadri. 2025. Elr-1000: A community-generated dataset for endangered indic indigenous languages. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2441–2457.

Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K Jain, Virginia Aglietti, Disha Jindal, Yuanzhu Peter Chen, et al. 2025. Big-bench extra hard. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26473–26501.

Supriya Khadka and Bijayan Bhattarai. 2025. Gender bias in nepali-english machine translation: A comparison of llms and existing mt systems. In *Proceedings of the 6th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 75–82.

Supriya Khadka, Ranju G.C., Prabin Paudel, Rahul Shah, and Basanta Joshi. 2023. Nepali text-to-speech synthesis using tacotron2 for mel-spectrogram generation. In *SIGUL 2023, 2nd Annual Meeting of the Special Interest Group on Under-resourced Languages: a Satellite Workshop of Interspeech 2023*.

Ju-Chun Ko. 2026. Bilingual bias in large language models: A taiwan sovereignty benchmark study. *arXiv preprint arXiv:2602.06371*.

Britt Krause. 1980. Kinship, hierarchy and equality in north western nepal. *Contributions to Indian Sociology*, 14(2):169–194.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Xingjun Ma, Yixu Wang, Hengyuan Xu, Yutao Wu, Yifan Ding, Yunhan Zhao, Zilong Wang, Jiabin

Hua, Ming Wen, Jianan Liu, Ranjie Duan, Yifeng Gao, Yingshui Tan, Yunhao Chen, Hui Xue, Xin Wang, Wei Cheng, Jingjing Chen, Zuxuan Wu, Bo Li, and Yu-Gang Jiang. 2026. A safety report on GPT-5.2, Gemini 3 Pro, Qwen3-VL, Grok 4.1 Fast, Nano Banana Pro, and Seedream 4.5. *arXiv preprint arXiv:2601.10527*.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. 2023. Aksharantar: Open indic-language transliteration datasets and models for the next billion users. In *Findings of the association for computational linguistics: Emnlp 2023*, pages 40–57.

Arijit Maji, Raghvendra Kumar, Akash Ghosh, Anushka Anushka, and Sriparna Saha. 2025. Sanskriti: A comprehensive benchmark for evaluating language models' knowledge of indian culture. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4434–4451.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. Stereoset: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 5356–5371.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1953–1967.

Jinu Nyachhyon, Mridul Sharma, Prajwal Thapa, and Bal Krishna Bal. 2025. Consolidating and developing benchmarking datasets for the nepali natural language understanding tasks. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1906–1925.

OpenRouter. n.d. Llm rankings. https://openrouter.ai/rankings. Accessed: 2026-02-21.

Shubham Parashar, Blake Olson, Sambhav Khurana, Eric Li, Hongyi Ling, James Caverlee, and Shuiwang Ji. 2025. Inference-time computations for llm reasoning and planning: A benchmark and insights. *arXiv preprint arXiv:2502.12521*.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. 2022. Bbq: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105.

Uwe Peters, Alexander Krauss, and Oliver Braganza. 2022. Generalization bias in science. *Cognitive science*, 46(9):e13188.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017.

Mengyao Piao, Lingqi Miao, Yilun Liu, Minggui He, Hongxia Ma, Li Zhang, Daimeng Wei, and Shimin Tao. Beyond the west: A survey of cultural datasets for culturally-grounded llms.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. Xcopa: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376.

Greeshma Prabha, PV Jyothsna, KK Shahina, B Premjith, and KP Soman. 2018. A deep learning approach for part-of-speech tagging in nepali language. In *2018 international conference on advances in computing, communications and informatics (ICACCI)*, pages 1132–1136. IEEE.

Ashish Pradhan and Archit Yajnik. 2024. Parts-of-speech tagging of nepali texts with bidirectional lstm, conditional random fields and hmm. *Multimedia Tools and Applications*, 83(4):9893–9909.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 4463–4473.

Tej Bahadur Shahi and Chiranjibi Sitaula. 2022. Natural language processing for nepali text: A review. *Artificial Intelligence Review*, 55(4):3401–3429.

Oyesh Mann Singh, Ankur Padia, and Anupam Joshi. 2019. Named entity recognition for nepali

language. In *2019 IEEE 5th international conference on collaboration and internet computing (cic)*, pages 184–190. IEEE.

Madhavendra Thakur. 2025. Culturally-grounded chain-of-thought (cg-cot): Enhancing llm performance on culturally-specific tasks in low-resource languages. *arXiv preprint arXiv:2506.01190*.

Surendrabikram Thapa, Kritesh Rauniyar, Hariram Veeramani, Surabhi Adhikari, Imran Razzak, and Usman Naseem. 2025. Probing the limits of multilingual language understanding: Low-resource language proverbs as llm benchmark for ai wisdom. In *Proceedings of the 6th Workshop on Computational Approaches to Discourse, Context and Document-Level Inferences (CODI 2025)*, pages 120–129.

Sulav Timilsina, Milan Gautam, and Binod Bhattarai. 2022. Nepberta: Nepali language model trained in a large corpus. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 273–284.

Armin Toroghi, Ali Pesaranghader, Tanmana Sadhu, and Scott Sanner. 2025. Llm-based typed hyperresolution for commonsense reasoning with knowledge bases. In *The Thirteenth International Conference on Learning Representations*.

Sshubam Verma, Mohammed Safi Ur Rahman Khan, Vishwajeet Kumar, Rudra Murthy, and Jaydeep Sen. 2025. Milu: A multi-task indic language understanding benchmark. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10076–10132.

Ieva Zake. 2023. Holistic bias in sociology: Contemporary trends. In *The Palgrave Handbook of Methodological Individualism: Volume II*, pages 403–421. Springer.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chengxing Xie, Cunxiang Wang, et al. 2026. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*.

P. Zhang, J. Wang, X. Hu, X. Wang, X. Fan, W. Chi, and W. Yang. 2026. Comparative performance of GPT-4, GPT-o3, GPT-5, Gemini-3-Flash, and DeepSeek-R1 in ophthalmology question answering. *Frontiers in Cell and Developmental Biology*, 14:1744389.

11. Appendix

Appendix A: Representative Examples from NeCCo

This appendix presents representative NeCCo examples in Devanagari Nepali, Romanized Nepali, and English (Table 3). For categories without official English translations (Kinship & Social Hierarchy; Idioms, Proverbs & Metaphors), we provide an English gloss for readability; this is not part of the benchmark.

Appendix B: Complete Results Across All Categories

This appendix reports per-model results across all NeCCo categories and input types. For *Proverbs/Idioms* and *Kinship & Social Hierarchy*, English translations are omitted to preserve meaning; corresponding columns are marked 0 (not evaluated). Across all tables (Table 4 – Table 9), **A** = accuracy, **P** = precision, **R** = recall, and **F1** = F1 score.

Appendix C: OpenRouter Usage Snapshot for Nepali

Figure 5 presents a snapshot of model usage statistics for the Nepali language on the OpenRouter platform, captured over a 30-day period ending in late March 2026. The data highlights Trinity Large Preview as the most frequently utilized model for Nepali prompts during this timeframe, accounting for 46.5% of the total volume.



Figure 5: Model usage share and token volume for the Nepali language on OpenRouter.

Category	Variant	Question	Options (A, B, C, D)	Ans.
Kinship & Social Hierarchy	Devanagari	बुवा को भिनाजु के हुन्छ?	A: आमाजू B: सालो C: छोरा D: फुपाजु	D
	Romanized	buwa ko bhinaju ke hunchha?	A: aamaju B: salo C: chhora D: phupaju	D
	English (gloss)	What do we call “father’s sister’s husband”?	A: aunt B: brother-in-law (wife’s brother) C: son D: paternal aunt’s husband	D
Festivals, Rituals & Geography	Devanagari	इन्द्रजात्रा मुख्यतः कहाँ मनाइन्छ?	A: पोखरा B: काठमाडौं C: भक्तपुर D: पाटन	B
	Romanized	Indrajatra mukhyata kaha manaincha?	A: Pokhara B: Kathmandu C: Bhaktapur D: Patan	B
	English	Where is Indra Jatra primarily celebrated?	A: Pokhara B: Kathmandu C: Bhaktapur D: Patan	B
Gastronomy, Agriculture & Nature	Devanagari	मास को दाल मा के झानिन्छ?	A: कागती B: मुला C: प्याज D: जिम्बु	D
	Romanized	maas ko daal ma ke jhanincha?	A: kagati B: mula C: pyaj D: jimbu	D
	English	What is typically tempered in black gram (maas) lentil soup?	A: lemon B: radish C: onion D: jimbu	D
Idioms, Proverbs & Metaphors	Devanagari	लुटको धन फुपुको श्राद्ध	A: पुण्य कमाउनु B: अरूको कमाइ खर्च गर्नु C: श्राद्ध गर्नु D: कमाउनु	B
	Romanized	lutko dhan phupuko shraddha	A: punya kamaunu B: aruko kamai kharcha garnu C: shraddha garnu D: kamaunu	B
	English (gloss)	Idiom meaning closest to:	A: earn merit B: spend someone else’s earnings C: perform a death ritual D: earn	B

Table 3: Representative NeCCo examples shown in parallel variants. English lines for Kinship and Idioms are explanatory glosses and are not part of the benchmark release.

Dataset	English				Devanagari				Romanized			
Model	A	P	R	F1	A	P	R	F1	A	P	R	F1
GPT-4o Mini	0.462	0.494	0.417	0.406	0.521	0.564	0.521	0.631	0.321	0.388	0.321	0.410
Trinity Large	0.280	0.282	0.280	0.320	0.369	0.435	0.39	0.422	242	0.287	0.242	0.215
DeepSeek-V3.2	0.351	0.370	0.351	0.466	0.399	0.422	0.399	0.000	0.242	0.276	0.242	0.000
Gemini 3 Flash	0.509	0.525	0.509	0.512	0.441	0.448	0.441	0.000	0.395	0.426	0.395	0.000
MiniMax-M2.5	0.333	0.365	0.333	0.000	0.399	0.412	0.366	0.000	0.322	0.370	0.322	0.000
GPT-5.2	0.548	0.568	0.548	0.000	0.530	0.563	0.530	0.000	0.266	0.327	0.266	0.202
GPT-OSS-120B	0.331	0.356	0.331	0.000	0.387	0.410	0.387	0.000	0.364	0.392	0.364	0.000
Grok-4.1 Fast	0.506	0.508	0.506	0.000	0.571	0.592	0.571	0.534	0.452	0.474	0.452	0.456
GLM-5	0.5952	0.614	0.595	0.000	0.595	0.614	0.595	0.000	0.436	0.488	0.436	0.000

Table 4: Performance comparison on the **Agriculture** category.

Dataset	English				Devanagari				Romanized			
Model	A	P	R	F1	A	P	R	F1	A	P	R	F1
GPT-4o Mini	0.840	0.842	0.840	0.824	0.870	0.920	0.870	0.887	0.764	0.871	0.764	0.803
Trinity Large	0.879	0.882	0.879	0.854	0.905	0.905	0.905	0.904	0.824	0.882	0.824	0.851
DeepSeek-V3.2	0.839	0.831	0.839	0.825	0.920	0.952	0.920	0.932	0.749	0.860	0.749	0.791
Gemini 3 Flash	0.864	0.868	0.864	0.847	0.980	0.980	0.980	0.980	0.968	0.976	0.970	0.972
MiniMax-M2.5	0.854	0.865	0.854	0.841	0.905	0.923	0.905	0.911	0.719	0.873	0.719	0.769
GPT-5.2	0.879	0.873	0.879	0.857	0.950	0.958	0.950	0.952	0.960	0.969	0.960	0.963
GPT-OSS-120B	0.819	0.836	0.819	0.816	0.867	0.938	0.867	0.893	0.677	0.835	0.677	0.737
Grok-4.1 Fast	0.844	0.854	0.844	0.823	0.935	0.949	0.935	0.940	0.859	0.916	0.859	0.882
GLM-5	0.859	0.845	.859	0.837	0.935	0.946	0.935	0.940	0.864	0.902	0.864	0.879

Table 5: Performance comparison on the **Cultural Etiquette** category.

Dataset	English				Devanagari				Romanized			
Model	A	P	R	F1	A	P	R	F1	A	P	R	F1
GPT-4o Mini					0.586	0.628	0.586	0.582	0.491	0.530	0.431	0.476
Trinity Large	0.000	0.000	0.000	0.000	0.453	0.485	0.453	0.423	0.464	0.494	0.464	0.4449
DeepSeek-V3.2	0.000	0.000	0.000	0.000	0.556	0.561	0.556	0.557	0.551	0.552	0.551	0.551
Gemini 3 Flash	0.000	0.000	0.000	0.000	0.827	0.831	0.827	0.828	0.833	0.837	0.833	0.834
MiniMax-M2.5	0.000	0.000	0.000	0.000	0.590	0.596	0.590	0.592	0.483	0.423	0.483	0.483
GPT-5.2	0.000	0.000	0.000	0.000	0.586	0.639	0.586	0.595	0.812	0.812	0.812	0.812
GPT-OSS-120B	0.000	0.000	0.000	0.000	0.532	0.542	0.532	0.535	0.513	0.516	0.513	0.513
Grok-4.1 Fast	0.000	0.000	0.000	0.000	0.714	0.726	0.714	0.715	0.697	0.710	0.697	0.697
GLM-5	0.000	0.000	0.000	0.000	0.701	0.707	0.701	0.701	0.718	0.722	0.718	0.718

Table 6: Performance comparison on the **Kinship & Social Hierarchy** category.

Dataset	English				Devanagari				Romanized			
Model	A	P	R	F1	A	P	R	F1	A	P	R	F1
GPT-4o Mini	0.000	0.000	0.000	0.000	0.742	0.786	0.742	0.759	0.500	0.619	0.500	0.533
Trinity Large	0.000	0.000	0.000	0.000	0.468	0.717	0.468	0.460	0.374	0.714	0.374	0.363
DeepSeek-V3.2	0.000	0.000	0.000	0.000	0.658	0.798	0.658	0.704	0.390	0.608	0.390	0.434
Gemini 3 Flash	0.000	0.000	0.000	0.000	0.878	0.903	0.878	0.888	0.862	0.877	0.862	0.869
MiniMax-M2.5	0.000	0.000	0.000	0.000	0.684	0.778	0.684	0.709	0.474	0.671	0.474	0.529
GPT-5.2	0.000	0.000	0.000	0.000	0.342	0.563	0.342	0.407	0.768	0.799	0.737	0.752
GPT-OSS-120B	0.000	0.000	0.000	0.000	0.614	0.715	0.614	0.647	0.471	0.656	0.471	0.517
Grok-4.1 Fast	0.000	0.000	0.000	0.000	0.837	0.882	0.837	0.851	0.700	0.796	0.700	0.739
GLM-5	0.000	0.000	0.000	0.000	0.847	0.885	0.847	0.860	0.711	0.817	0.711	0.755

Table 7: Performance comparison on the **Proverbs/Idioms** category.

Dataset	English				Devanagari				Romanized			
Model	A	P	R	F1	A	P	R	F1	A	P	R	F1
GPT-4o Mini	0.516	0.480	0.516	0.474	0.722	0.722	0.722	0.720	0.538	0.51	0.538	0.503
Trinity Large	0.538	0.581	0.538	0.503	0.605	0.600	0.605	0.571	0.000	0.000	0.000	0.000
DeepSeek-V3.2	0.659	0.659	0.659	0.656	0.713	0.708	0.713	0.707	0.000	0.000	0.000	0.000
Gemini 3 Flash	0.842	0.842	0.842	0.841	0.837	0.838	0.837	0.836	0.000	0.000	0.000	0.000
MiniMax-M2.5	0.552	0.540	0.552	0.531	0.718	0.714	0.718	0.708	0.000	0.000	0.000	0.000
GPT-5.2	0.839	0.839	0.839	0.838	0.892	0.893	0.832	0.891	0.000	0.000	0.000	0.000
GPT-OSS-120B	0.572	0.550	0.572	0.547	0.649	0.623	0.649	0.619	0.534	0.505	0.534	0.000
Grok-4.1 Fast	0.7400	0.719	0.7400	0.710	0.749	0.738	0.749	0.000	0.000	0.000	0.000	0.000
GLM-5	0.753	0.745	0.753	0.000	0.753	0.745	0.753	0.000	0.000	0.000	0.000	0.000

Table 8: Performance comparison on the **Festivals** category.

Dataset	English				Devanagari				Romanized			
Model	A	P	R	F1	A	P	R	F1	A	P	R	F1
GPT-4o Mini	0.470	0.475	0.470	0.000	0.633	0.628	0.633	0.627	0.412	0.422	0.412	0.413
Trinity Large	0.251	0.521	0.261	0.228	0.369	0.588	0.369	0.362	0.190	0.455	0.180	0.154
DeepSeek-V3.2	0.378	0.470	0.380	0.401	0.516	0.599	0.516	0.536	0.259	0.447	0.259	0.282
Gemini 3 Flash	0.612	0.632	0.612	0.617	0.610	0.614	0.608	0.610	0.497	0.496	0.497	0.494
MiniMax-M2.5	0.456	0.497	0.455	0.470	0.556	0.608	0.556	0.572	0.387	0.484	0.387	0.413
GPT-5.2	0.634	0.671	0.634	0.646	0.778	0.784	0.778	0.766	0.655	0.636	0.655	0.613
GPT-OSS-120B	0.514	0.550	0.514	0.526	0.586	0.640	0.586	0.602	0.540	0.575	0.540	0.551
Grok-4.1 Fast	0.520	0.546	0.520	0.527	0.620	0.635	0.620	0.625	0.420	0.472	0.420	0.438
GLM-5	0.617	0.636	0.617	0.623	0.617	0.636	0.617	0.623	0.425	0471	0.425	0.440

Table 9: Performance comparison on the **Geography** category.

Reward-Guided Fine-Tuning of Whisper for Low-Resource Nepali Speech Recognition

Aadarsh Pandit, Yudhin Khanal, Ishan Pandey, Kushal Kunwar, Sunil Regmi
aadarshapandit@gmail.com, yudhinkhanal548@gmail.com, eshanpandey804@gmail.com,
kunwarkushal190@gmail.com, sunil.regmi@ku.edu.np
Department of Artificial Intelligence, Kathmandu University, Dhulikhel, Nepal

Abstract

Fine-tuning speech recognition models on noisy real-world data is tricky. The model has no way of knowing which training samples are reliable and which are not, so it ends up learning from bad examples just as readily as good ones. This is a real problem for Nepali, where most available training data comes from YouTube videos with automatically generated subtitles that are often inaccurate. In this work, we tried a simple fix. Instead of feeding everything to the model, we first asked humans to rate the quality of a sample of transcriptions, trained a small Random Forest classifier on those 2,000 ratings, and used it to filter out the bad samples before each retraining round. The classifier uses four automatically computable features, Word Error Rate (WER), Character Error Rate (CER), length ratio, and length difference, and achieves 81% accuracy on a held-out set. Running two filtering and retraining cycles on a 40,000 clip training subset drawn from a 68.4 hour corpus improves substantially over our own standard fine-tuning baseline of 5.60% WER and 5.10% CER, reaching 4.89% WER and 4.52% CER, which corresponds to an 11–13% relative gain. The approach is much lighter than full Reinforcement Learning from Human Feedback (RLHF) but still uses real human judgment to guide training.

Keywords: Low-Resource ASR, Reward Model, Data Filtering, Whisper, Nepali, Human Feedback

1. Introduction

Automatic Speech Recognition has improved a lot in recent years, largely because of large pretrained models like wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and Whisper (Radford et al., 2023). These models work surprisingly well across many languages out of the box, which makes them a natural starting point when you want to build ASR for a language that does not have much data. The usual approach is to just take one of these pretrained models and fine-tune it on whatever local data you can find.

Nepali is a morphologically rich South Asian language written in Devanagari, an abugida script used across South Asia for languages including Hindi, Sanskrit, and Marathi, and like many South Asian languages it remains severely underresourced for speech technology despite having tens of millions of speakers. For Nepali, the most accessible source of training data is YouTube, specifically audio clips paired with SRT (SubRip Subtitle) files. The problem is that the subtitle quality varies a lot. Some are accurate, many are not, and when you fine-tune a model on all of it equally, it learns from the bad examples just as much as the good ones and gradually gets worse at the things it was already getting wrong.

Prior work has tried to get around this. Ghimire et al. (2023) used language model scores to pick which pseudo-labeled samples to keep, reaching 6.77% CER on the OpenSLR Nepali test set. But Language Model (LM) scores measure whether the output sounds fluent, not whether it actually matches what was said. Pseudo-labeling methods (Singh et al., 2023; Xu et al., 2020) face the same limitation. They rely entirely on the model's own confidence.

Our approach is different: we ask humans. We built a small annotation tool, collected 2,000 quality ratings on reference-hypothesis pairs, and trained a simple Random Forest classifier to predict those ratings from four error metrics. That classifier then acts as a filter. Before each retraining round, it removes the samples the model is most likely to get wrong. The result is a progressively cleaner training set and steadily improving accuracy, without any of the complexity of full RLHF.

Our main contributions are:

- A browser-based annotation platform for rating Nepali ASR transcription quality.
- A Random Forest reward model using four automatic features, achieving 81% accuracy.
- An iterative filtering pipeline that improves WER from 5.60% to 4.89% and CER from 5.10% to 4.52%, an 11–13% relative gain over our own standard fine-tuning baseline.
- Results on a 68.4-hour real-world Nepali speech corpus covering news, podcasts, and serials.

2. Related Work

2.1. Nepali ASR

Early Nepali ASR work relied on classical architectures. Regmi et al. (2019) built an RNN-CTC system, while Regmi and Bal (2021) pushed further

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 169–175
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

with a joint CTC-attention model that dropped the need for a pronunciation lexicon, reporting 0.30% CER on the controlled OpenSLR test set. CNN-based approaches from Banjara et al. (2020) and Dhakal et al. (2022) trailed behind at 27.72% and 17.07% CER respectively. More recently, Rijal et al. (2024) showed that fine-tuning OpenAI Whisper on Nepali data yields strong gains, and Ghimire et al. (2023) used language model scores to iteratively select pseudo-labeled samples for MMS-1B (Pratap et al., 2023), reaching 6.77% CER.

2.2. Active Learning and Self-Training

Several groups have explored using the model's own outputs to drive training. Zheng et al. (2023) used HuBERT-based perplexity scores for contrastive data selection, cutting WER by 11%. Singh et al. (2023) iterated on pseudo-labels above a confidence threshold with wav2vec 2.0 XLSR-53. Xu et al. (2020) showed iterative pseudo-labeling helps on LibriSpeech, and Xu et al. (2021) confirmed self-training and pretraining are complementary. All of these approaches share a limitation: the selection signal comes from the model itself, not from external human judgment.

2.3. Reward Models and Human Feedback

RLHF has been highly effective for aligning language models (Ouyang et al., 2022), but applying it directly to encoder-decoder ASR systems is cumbersome. It requires running multiple model copies, managing KL (Kullback-Leibler divergence) constraints, and keeping policy gradient training stable. We take a lighter-weight approach: train a reward classifier from human annotations and use it purely as a data filter, keeping all the benefit of human-grounded quality signals without the Reinforcement Learning (RL) machinery.

3. Methodology

Our pipeline has three stages: fine-tune Whisper Small on the 40,000-clip training pool to get a baseline model, train a reward classifier on human-annotated quality scores, then iteratively filter the training pool and retrain. We use the term *reward model* analogously to its use in RLHF: a function trained on human judgments that assigns a scalar quality score to a model output. Unlike full RLHF, however, we use this score purely as a data filter rather than as a gradient signal. Figure 1 shows the overall flow.

3.1. Baseline Fine-Tuning

We start from OpenAI's Whisper Small (Radford et al., 2023), a 244M-parameter transformer encoder-decoder pretrained on multilingual audio. Audio goes through a convolutional frontend that produces log-mel spectrograms, which the encoder processes into acoustic representations; the decoder then generates text conditioned on those representations and task tokens.

Fine-tuning uses `WhisperForConditionalGeneration` and `WhisperProcessor` from Hugging Face Transformers, with a custom data collator that pads sequences dynamically and masks padding positions from the loss using -100. We train for 3 epochs with batch size 4, gradient accumulation over 2 steps, FP16 precision, and checkpoint saves every 500 steps. This produces FT_{model_0}.

3.2. Reward Model

3.2.1. Annotation Platform

To get human quality judgments, we built a small Flask-based annotation tool. We ran FT_{model_0} on audio from OpenSLR54 (Kjartansson et al., 2018) to generate hypothesis transcriptions, then presented each annotator with the reference text and the model's output side by side. They could play the audio and assign one of three labels: **1 (Good)** for accurate output with minimal errors, **0 (Neutral)** for output that is understandable but noticeably imperfect, and **-1 (Bad)** for output with major errors or hallucinations. We collected exactly 2,000 labeled pairs this way. Annotation was performed by five members of the research team, with each sample assigned to one annotator. To verify consistency, 200 samples (10%) were independently labeled by a second annotator; the two annotators agreed on 83% of these, with disagreements concentrated at the Good/Neutral boundary rather than the Bad category, suggesting the Bad label which drives all filtering decisions is applied reliably.

3.2.2. Feature Engineering

Each (reference, hypothesis) pair is described by four features: WER (word-level edit distance), CER (character-level edit distance, which matters a lot for morphologically rich Devanagari), length_ratio (hypothesis word count divided by reference word count), and length_diff (absolute word count difference). These four numbers can be computed automatically from any reference-hypothesis pair without needing extra linguistic tools.

3.2.3. Reward Model Training

We train a `RandomForestClassifier` on the 2,000 annotated pairs using an 80/20 split. The model predicts the three-class quality label and

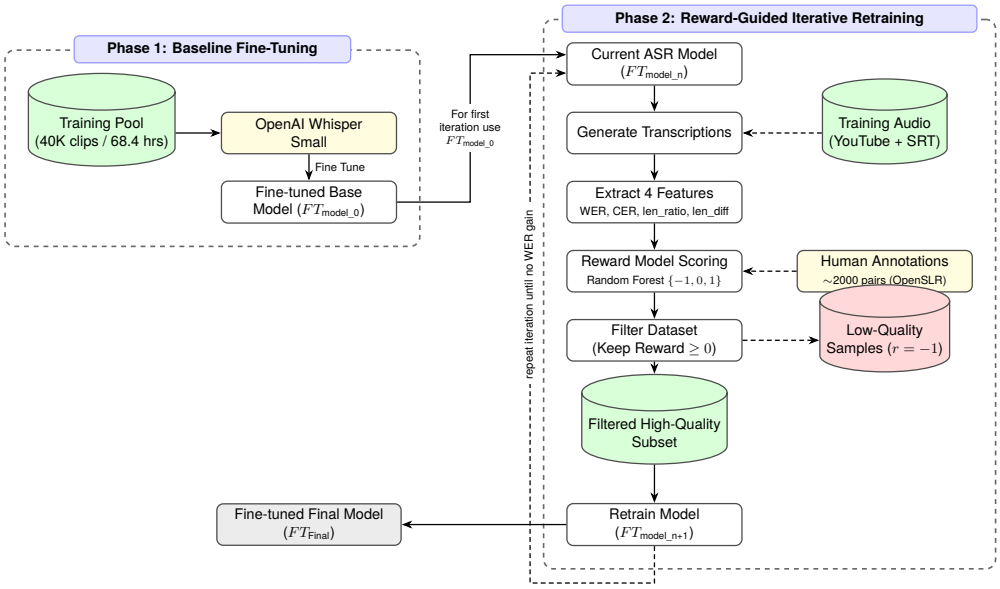


Figure 1: Architecture of the reward-guided fine-tuning pipeline.

reaches **81% accuracy** on the held-out set. A natural question is why a classifier is needed at all rather than a simple WER or CER threshold. The reason is that human quality judgments do not reduce cleanly to a single metric: annotators frequently rated samples as Bad despite low WER (e.g., when the hypothesis was fluent but semantically wrong), and as Good despite moderate WER (e.g., when the reference SRT itself contained a minor typo). A fixed threshold on any single metric cannot capture these interactions; the Random Forest learns them from the annotated data. We also evaluated a Multi-Layer Perceptron neural network as an alternative, but it overfit severely on the 2,000-sample training set. Random Forest was selected because it handles nonlinear feature interactions well, is robust to the outlier WER/CER values that come from very short clips, and achieved the best held-out validation accuracy. We used default scikit-learn hyperparameters (100 trees, `max_features='sqrt'`) without further tuning, which proved sufficient. Feature importance analysis put WER and CER together at about 65% of total importance, with length features covering the rest. The length_diff feature in particular catches hallucinations that error rates miss when the reference is very short. We keep samples with predicted reward $r \geq 0$ (Good or Neutral) and drop only the Bad ones; removing Neutral samples too would shrink the training set too aggressively.

3.3. Reward-Guided Iterative Filtering

Algorithm 1 shows the full procedure. After the baseline fine-tuning step, we run the current model over the 40,000-clip training pool, compute the four features for each sample, and ask the reward model whether to keep it. Samples predicted as Bad ($r = -1$) are dropped; the rest go into the next round of fine-tuning. We repeat until WER stops improving on the held-out test set.

It is important to note the two-stage nature of the feature computation. During reward model *training*, WER and CER are computed against the clean, human-verified OpenSLR54 transcriptions, so the classifier learns from high-quality reference signal. During reward model *inference* on the YouTube training pool, however, the only available reference is the noisy SRT subtitle. This means that in cases where the model produces a correct transcription but the SRT is wrong, the computed WER will be artificially high and the reward model may incorrectly mark the sample as Bad. We acknowledge this as a limitation: the filter is bounded by the quality of the SRT proxy reference. In practice, the consistent WER improvements across iterations suggest that correctly-transcribed samples with bad SRTs are a small minority and do not dominate the filtering signal, but this assumption would benefit from explicit verification in future work.

The key difference from Ghimire et al. (2023) is what the filter actually measures. Their LM score tells you whether the hypothesis sounds like fluent Nepali. Ours tells you whether it matches the reference. That is a more direct proxy for transcription quality, and it does not require maintaining a separate language model.

4. Dataset

Our training corpus consists of Nepali audio collected from YouTube: news programs, podcasts,

Algorithm 1 Reward-Guided Fine-Tuning

Require: Pretrained model M_0, training dataset D, reward model R, test dataset D_{test}
Ensure: Final fine-tuned model M_{final}

$M_{final} \leftarrow \text{FineTune}(M_0,\ D)$
repeat
  $D_{filtered} \leftarrow \{\}$
  for each $(x,\ y)$ **in** D **do**
    $\hat{y} \leftarrow M_{final}.\text{Transcribe}(x)$
    $F \leftarrow \text{features}(\hat{y},\ y)$ ▷ WER, CER, len_ratio, len_diff
    $r \leftarrow R.\text{Predict}(F)$
    if $r \geq 0$ **then**
      $D_{filtered}.\text{Append}((x,\ y))$
    end if
  end for
  $M_{final} \leftarrow \text{FineTune}(M_{final},\ D_{filtered})$
  $\text{WER} \leftarrow \text{ComputeWER}(M_{final},\ D_{test})$
until WER does not improve
return M_{final}

drama serials, and documentaries. We downloaded SRT subtitle files alongside the videos and used ffmpeg to cut the audio at subtitle boundaries, then resampled everything to 16 kHz. Table 1 summarizes the corpus statistics.

Property	Value
Total audio segments	49,216
Total duration	68.4 hours
Dataset size	18.9 GB
Training pool	40,000 clips
Validation/Test set	9,216 clips
Average clip length	∼6 seconds
Sampling rate	16 kHz
Sources	News, podcasts, serials, docs
Annotation samples (OpenSLR)	2,000

Table 1: Training dataset statistics.

4.1. Data Processing

Raw SRT files were parsed to extract timestamp-text pairs, and ffmpeg was used to slice the corresponding audio segments. Clips shorter than 1 second or longer than 30 seconds were discarded to avoid uninformative or over-segmented samples. All audio was converted to mono, resampled to 16 kHz, and saved as WAV files. From the resulting 49,216 clips, we randomly partitioned the data into a training pool of 40,000 clips and a held-out evaluation set of 9,216 clips (split between validation and testing). Text transcriptions were normalized by removing special characters, extra whitespace, and non-Nepali tokens.

For the reward model annotations, we drew from OpenSLR54 (Kjartansson et al., 2018), a crowd-sourced Nepali corpus with 165 hours from 527 speakers. Human annotators labeled 2,000 reference-hypothesis pairs; the distribution came out at approximately 38% Good, 29% Neutral, and 33% Bad, confirming that a meaningful fraction of the model's outputs are noisy enough to hurt retraining if left in.

5. Experiments and Results

5.1. Experimental Setup

All experiments ran on an NVIDIA RTX 4060 GPU paired with an AMD Ryzen 7 7000-series CPU and 16 GB RAM. The reward model was trained with `scikit-learn`, and WER/CER were computed with the `jiwer` library. All models share the same held-out test split.

5.2. Quantitative Results

Table 2 shows performance at each stage. We first fine-tuned Whisper Small on our 40,000-clip training pool for 3 epochs to produce FT_{model_0}, our own baseline, which achieved 5.60% WER and 5.10% CER (row 2). This serves as the direct starting point for all reward-guided iterations and is the model against which percentage improvements are measured. For reference we also include the published Whisper FT result of Rijal et al. (2024) on Nepali (row 3), which used the same architecture on different data. Applying our reward-guided iterative filtering to the training pool consistently improves on our own baseline. Each filtering pass removed approximately 30–35% of that round's available samples: Iteration 1 retained 27,200 clips from the 40,000-clip pool (removing 32%), and Iteration 2 retained 18,500 clips from the 27,200 remaining after Iteration 1 (removing 32%). The final model reaches 4.89% WER and 4.52% CER, an 11–13% relative reduction over the baseline.

Training loss dropped steadily across all stages: from 1.426 to 0.052 during the baseline fine-tuning run, then from 0.052 to 0.031 after Reward-Guided Iteration 1, and from 0.031 to 0.024 after Iteration 2. This progressive reduction reflects a cleaner, less contradictory training signal at each round rather than the task simply becoming easier due to dataset shrinkage.

5.3. Comparison with Prior Work

Table 3 compares our results with previous Nepali ASR systems. The reported numbers are taken directly from the respective papers, and it is important to note that they were evaluated on different test sets; we did not re-run these systems on our dataset.

Training Stage	WER (%)	CER (%)
Base Whisper (no fine-tuning)	8.10	7.40
Our Baseline FT, 40K clips (this work)	5.60	5.10
Whisper FT (Rijal et al., 2024) (reference)	—	5.04
Reward-Guided Iteration 1 (27,200 clips)	5.12	4.71
Reward-Guided Iteration 2 (18,500 clips)	**4.89**	**4.52**

Table 2: WER and CER across training stages. Row 2 is our own baseline fine-tuning of Whisper Small on the 40,000-clip training pool; this is the direct starting point for all reward-guided iterations. Row 3 shows the published Rijal et al. result for architectural reference; WER was not reported in that work (—). All rows from Base Whisper onward are evaluated on the same held-out test split.

Among systems evaluated on diverse real-world Nepali audio, excluding the controlled OpenSLR result of Regmi and Bal (2021), our model achieves the lowest reported CER at 4.52%. We include Rijal et al. (2024) primarily as an architectural reference point, and its result is also presented in Table 2 as a baseline.

The 0.30% CER reported by Regmi and Bal (2021) was obtained on the clean and controlled OpenSLR test split, and is therefore not directly comparable to evaluations on real-world data.

Method	Architecture	CER (%)
Banjara et al. (2020)	CNN-GRU	27.72
Dhakal et al. (2022)	CNN-BiLSTM	17.07
Regmi and Bal (2021)	CTC-Attention	0.30†
Ghimire et al. (2023)	MMS-1B + Active LM	6.77
Rijal et al. (2024)	Whisper FT	5.04
Proposed	**Reward-Guided Whisper**	**4.52**

Table 3: Comparison with prior Nepali ASR systems. †Evaluated on a controlled OpenSLR test split; not directly comparable to our real-world corpus.

5.4. Qualitative Analysis

Table 4 shows a few examples comparing outputs from the baseline fine-tuned model (FT_{model_0}) and the reward-guided final model. The baseline fine-tuning makes repeatable errors: dropped syllables in compound words, wrong morphological endings, incorrect word boundaries (e.g., splitting *Nepalako* into *Nepal ko*). Reward-guided retraining tends to get these right, most likely because the bad training samples responsible for those errors have been filtered out.

Reference	Baseline FT	Reward-Guided
नेपालको राजधानी काठमाडौं हो।	नेपाल को रजधान काटमाण्डु हो।	नेपालको राजधानी काठमाडौं हो।
आज मौसम निकै सफा छ।	आज मसम ननको साफ छ।	आज मौसम निकै सफा छ।
हामी अहिले बैठकमा छौं।	हामी अहिले बैठमा छौ।	हामी अहिले बैठकमा छौं।

Table 4: Sample transcription comparisons.

6. Conclusion

We set out to tackle a straightforward problem: when you fine-tune a speech model on noisy YouTube data, it learns from bad examples as readily as good ones. Our solution was to bring in human judgment, not through the full machinery of RLHF, but through a simple classifier trained on 2,000 annotated transcription pairs. Two rounds of reward-guided filtering and retraining improved WER from 5.60% to 4.89% and CER from 5.10% to 4.52%, an 11–13% relative gain over our own standard fine-tuning baseline, running on a single consumer GPU. Looking ahead, more annotations would improve reward model coverage, neural alternatives to Random Forest could capture richer quality signals, and integrating language model rescoring at decode time could squeeze out further gains. The same pipeline should transfer naturally to other low-resource South Asian languages facing the same noisy subtitle problem.

Ethical Considerations

This work involves human annotation of speech transcription quality. Annotators participated voluntarily and were members of the research team. No personally identifiable information was collected during annotation. The training data was sourced from publicly available YouTube content; we do not redistribute the raw audio. The annotation labels and extracted features used to train the reward model do not contain sensitive information.

Limitations

The reward model is trained on only 2,000 annotated samples, which limits its ability to capture the full diversity of Nepali speech, including variations in accent, speaking style, and domain. The four hand-crafted features (WER, CER,

length_ratio, and length_diff) are useful but do not cover all aspects of transcription quality, especially errors related to meaning rather than surface form.

Another important limitation comes from how features are computed during inference. When applied to the YouTube training pool, WER and CER are calculated against noisy SRT subtitles rather than clean references. As a result, the filter may discard cases where the model produces a correct transcription but the subtitle is incorrect, effectively penalizing the model for fixing errors in the data. Although the consistent improvements across iterations suggest that such cases are relatively uncommon, this remains a fundamental limitation of the approach. Fully addressing it would require access to clean, human-verified references at scale.

Finally, our experiments were conducted on a single GPU setup, and the iterative retraining process can become computationally expensive when scaled to larger datasets or more iterations.

Data and Code Availability

The Nepali ASR audio dataset (49,216 clips, ∼68.4 hours, 18.9 GB) is publicly available on Hugging Face at https://huggingface.co/datasets/Aadarsh17/nepali-asr-dataset-public under a CC BY-NC 4.0 licence. The reward model code and full pipeline are available on GitHub at https://github.com/Aadarshttech/nepali-asr-reward-guided. The OpenSLR54 corpus used for annotation is publicly available at https://www.openslr.org/54/.

Acknowledgments

The authors thank the members of the Department of Artificial Intelligence at Kathmandu University for their support during annotation and evaluation.

Bibliographical References

A. Baevski, H. Zhou, A. Mohamed, and M. Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates Inc.

J. Banjara, K. R. Mishra, J. Rathi, K. Karki, and S. Shakya. 2020. Nepali speech recognition using CNN and sequence models. In *2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, pages 1–5. IEEE.

M. Dhakal, A. Chhetri, A. K. Gupta, P. Lamichhane, S. Pandey, and S. Shakya. 2022. Automatic speech recognition for the Nepali language using CNN, bidirectional LSTM and ResNet. In *2022 International Conference on Inventive Computation Technologies (ICICT)*, pages 515–521. IEEE.

R. R. Ghimire, B. K. Bal, and P. Poudyal. 2023. Active learning approach for fine-tuning pre-trained ASR model for a low-resourced language. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON 2023)*, pages 82–89. NLP Association of India.

W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

O. Kjartansson, S. Sarin, K. Pipatsrisawat, M. Jansche, and L. Ha. 2018. Crowd-sourced speech corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In *Proceedings of the 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 52–55. ISCA.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates Inc.

V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, A. Baevski, Y. Adi, X. Zhang, W.-N. Hsu, A. Conneau, and M. Auli. 2023. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, volume 202, pages 28492–28518. PMLR.

P. Regmi, A. Dahal, and B. Joshi. 2019. Nepali speech recognition using RNN-CTC model. *International Journal of Computer Applications*, 178(31):1–6.

S. Regmi and B. K. Bal. 2021. An end-to-end speech recognition for the Nepali language. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON 2021)*, pages 180–185. NLP Association of India.

S. Rijal, S. Adhikari, M. Dahal, M. Awale, and V. Ojha. 2024. Whisper fine-tuning on Nepali language. *arXiv preprint arXiv:2411.12587*.

S. Singh, F. Hou, and R. Wang. 2023. A novel self-training approach for low-resource speech recognition. In *Proceedings of INTERSPEECH 2023*, pages 1588–1592. ISCA.

Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert. 2020. Iterative pseudo-labeling for speech recognition. In *Proceedings of INTERSPEECH 2020*, pages 1006–1010. ISCA.

Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, and M. Auli. 2021. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021 – IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3030–3034. IEEE.

Z. Zheng, Z. Ma, Y. Wang, and X. Chen. 2023. Unsupervised active learning: Optimizing labeling cost-effectiveness for automatic speech recognition. *arXiv preprint arXiv:2308.14814*.

Language Resource References

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. OpenSLR Nepali Speech Dataset (SLR54). In *Proceedings of the 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 52–55. ISCA. https://www.openslr.org/54/

Evaluating Linguistic Knowledge of LLMs in Tamil: The *ILAKKANAM* Benchmark

Jeyarajalingam Varsha[1], Menan Velayuthan[2], Sumirtha Karunakaran[1], Rasan Nivethiga[3], Kengatharaiyer Sarveswaran[1]
[1]Department of Computer Science, University of Jaffna, Sri Lanka
[2]Utrecht University, The Netherlands
[3]Department of Linguistics and English, University of Jaffna, Sri Lanka
varshajeyaraj@gmail.com, m.velayuthan@uu.nl, sumirthakarunakaran96@gmail.com, rnivethiga@univ.jfn.ac.lk, sarves@univ.jfn.ac.lk

Abstract

Large Language Models (LLMs) have shown strong generalization across tasks in high-resource languages; however, their linguistic competence in low-resource and morphologically rich languages such as Tamil remains largely unexplored. Existing multilingual benchmarks often rely on translated English datasets, failing to capture the language specific linguistic and cultural nuances of the target language. To address this gap, we introduce *ILAKKANAM*, the first Tamil-specific linguistic evaluation benchmark manually curated using 820 questions from Sri Lankan school-level Tamil subject examination papers spanning Grades 1–13. Each question is annotated by trained linguists under five linguistic categories and a factual knowledge category. We evaluate both closed-source and open-source LLMs using a standardized evaluation pipeline. Our results show that Gemini 2.5 achieves the highest overall performance, while open-source models lag behind, highlighting the gap in linguistic grounding. Category- and grade-wise analyses reveal that all models perform well on lower-grade questions but show a clear decline as the grade level and the linguistic complexity of the questions increase. Further, no strong correlation is observed between a model's overall performance and its ability to identify linguistic categories, suggesting that performance may be driven by exposure rather than genuine understanding. The code and dataset used in this study are publicly available in our repository, where the dataset consists only of extracted examination questions to mitigate potential data leakage.

Keywords: Tamil, Linguistic Benchmark, Linguistic diagnostics, Low-resource language

1. Introduction

Since the public release of ChatGPT in 2022 (OpenAI, 2022), Large Language Models (LLMs) have drawn significant public attention and rapidly integrated into everyday life. This growing interest has attracted substantial investment and funding toward companies developing these systems, resulting in a proliferation of models from different vendors. Closed-source models such as GPT-5 (OpenAI, 2025), Claude Sonnet 4.5 (Claude, 2025), and Gemini 2.5 (Comanici et al., 2025), as well as open-source counterparts like LLaMA 4 (Llama4Herd, 2025), DeepSeek-V3 (DeepSeek-AI et al., 2025), Qwen 2.5 (Qwen et al., 2025), and Grok 4 (Grok4, 2025), represent this expanding ecosystem. As LLMs become integrated into human workflows, including their use as evaluators for complex tasks (LLM-as-a-Judge) (Gu et al., 2025; Fu and Liu, 2025), the responsibility lies with the research community to evaluate these models, understand their capabilities, and identify their limitations (Chang et al., 2023).

The GLUE benchmark (Wang et al., 2018) and its extended version, SuperGLUE (Wang et al., 2019), established a standardized framework for evaluating language understanding across lexical semantics, logic, and grammar. BLiMP (Warstadt et al., 2020) extended this direction by introducing minimal pair evaluations for core syntactic phenomena such as subject–verb agreement and filler–gap dependencies. HELM (Liang et al., 2023) broadened the evaluation scope to incorporate ethical and demographic dimensions while emphasizing the limited multilingual and typological coverage of existing benchmarks. The MMLU dataset (Hendrycks et al., 2021) introduced a multitask evaluation across 57 academic and professional subject domains assessing both factual knowledge and reasoning ability. Despite measurable gains from larger models such as GPT-3, performance remains below expert level, indicating persistent limitations in knowledge depth and reliability.

While several efforts have attempted to create multilingual benchmarks, most are direct translations of their English counterparts (Singh et al., 2025; Bandarkar et al., 2024). Such approaches often fail to capture the cultural and linguistic nu-

The name ILAKKANAM (இலக்கணம்), meaning "grammar" or "structural rules" in Tamil, reflects the dataset's focus on evaluating the formal linguistic accuracy and systematic rule-adherence of large language models.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 176–189
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

ances of the target languages (Ji et al., 2023). Following the design of MMLU, comparable benchmarks have been developed for other languages, including Sinhala (Pramodya et al., 2025), Arabic (Koto et al., 2024), Chinese (Li et al., 2024), Turkish (Yüksel et al., 2024), Indonesian (Koto et al., 2023), Korean (Son et al., 2025), and Persian (Ghahroodi et al., 2024). These efforts demonstrate the importance of language-specific benchmarks developed by native-speaking communities, ensuring that linguistic and cultural characteristics are represented, which are otherwise often absent in large-scale multilingual settings. While SEA-HELM[2] (Susanto et al., 2025) evaluates the linguistic capabilities of LLMs through its LINDSEA suite for Tamil, its coverage remains limited.

To address this gap, we introduce the *ILAKKANAM*, a manually curated dataset designed for Tamil linguistic assessment. Inspired by MMLU, we compile questions from Sri Lankan school-level Tamil language examination papers.

ILAKKANAM comprises 820 questions spanning Grades 1–13, each annotated by trained linguists under five main linguistic categories (phonetics, phonology, morphology, syntax, and sematics). This paper outlines the procedures used to collect, clean, annotate, and evaluate these questions against both closed- and open-source LLMs. To summarize, our work makes the following core contributions:

- We introduce *ILAKKANAM*, a manually curated Tamil linguistic benchmark consisting of 820 questions from Sri Lankan school-level Tamil language examination papers, annotated across five linguistic categories and a factual knowledge category.
- We design a structured evaluation pipeline to assess both closed-source and open-source LLMs, enabling fine-grained comparison across linguistic dimensions and grade-level complexity.
- Through quantitative analysis, we reveal a consistent performance gap between closed- and open-source models, limited correlation between linguistic accuracy and category classification, and highlight the need for deeper linguistic grounding in Tamil language modeling.

The code and dataset are available online[3]. The dataset includes only extracted examination questions to minimise potential data leakage.

[2]Previously known as BHASHA

[3]https://github.com/LTG-UoJ/ILAKKANAM-Public

2. Background

This section provides a brief introduction to the Tamil language and the Sri Lankan education system to contextualise and support the work presented in this paper.

2.1. The Tamil language

Tamil (tam) is a member of the South Dravidian branch of the Dravidian language family and is spoken by approximately 90 million people worldwide[4]. It is an agglutinative language with rich morphological and syntactic constructions. Tamil has a documented history of over two millennia, evolving through distinct historical stages. It holds official language status in Sri Lanka, Singapore, and the Indian state of Tamil Nadu, and recognised as a second language in many other countries.

From a computational perspective, Tamil is classified as a low-resource language according to Joshi's typology (Joshi et al., 2020). Although widely spoken, Tamil lacks comprehensive, high-quality, and error-free resources—particularly annotated datasets and standardized benchmarks—which limits the development and evaluation of robust NLP tools for the language.

2.2. The Sri Lankan education system

Sri Lanka provides 13 years of free general education across four cycles: Primary (Grades 1–5, ages 5–10), Junior Secondary (Grades 6–9, ages 11–14), Senior Secondary (Grades 10–11, ages 15–16), and Advanced Level (Grades 12–13, ages 17–18)(Liyanage, 2014). Schooling is compulsory from Grade 1 to 13, with three major national examinations: the Grade 5 Scholarship Exam (for merit-based school access and financial aid), the GCE O/L at Grade 11 (stream selection for A/L), and the GCE A/L at Grade 13 (university admission)(Liyanage, 2014).

Tamil is taught as a subject for Tamil-speaking students from Grade 1 to 11, and as a specialization in Grades 12–13. This extended exposure, reinforced by national assessments, builds strong linguistic competence among Tamil speaking population in Sri Lanka.

By Grade 5[5], students are expected to acquire foundational skills in listening, reading, speaking, and writing. By Grade 11[6], they are expected to master grammar and apply Tamil in academic and

[4]https://www.worlddata.info/languages/tamil.php

[5]https://nie.lk/pdffiles/tg/tGR05TGTAMILLANGUAGE.pdf

[6]https://nie.lk/pdffiles/tg/t11tim159.pdf

social contexts. Though not always explicitly labeled, the curriculum gradually introduces key linguistic domains: phonetics, phonology, morphology, syntax, and semantics. In Grades 12–13, students specializing in Tamil engage with advanced material, including literature, history, and poetry.

3. Related Work

3.1. Tamil in Multilingual Benchmarks

A few multilingual benchmark evaluation datasets include Tamil among other Indic and Southeast Asian languages.

SEA-HELM (Susanto et al., 2025; Leong et al., 2023), through its LINDSEA suite for Tamil, assesses morphology, syntax and semantics using minimal pair structures and handcrafted diagnostics. In their work, the authors show that both GPT-4 and GPT-3.5-Turbo perform poorly on Tamil morphological analysis, with scores of 16.43% and 41.43%, respectively, even when prompted in English. Issues were particularly acute in processing gender, person, and tense agreement, as well as case marking in Question Answering tasks. This highlights that even prominent commercial models still have progress to make in achieving true multilinguality. SEA-HELM further revealed that Tamil's non-Latin script introduces complications in prompt design, such as the use of capitalization or punctuation conventions that do not align with Tamil's orthographic norms.

While PARIKSHA (Watts et al., 2024) scored Tamil-generated content highly in terms of fluency and grammaticality, it remains unclear whether this reflects deep language modeling or surface-level token fluency. Collectively, these studies provide strong evidence for the creation and evaluation of LLMs on Tamil in a more fine-grained manner.

3.2. Categorization of Linguistic Evaluation

Linguistic evaluation can span several axes. However, we use the five key levels: Phonetics, Phonology, Morphology, Syntax and Semantics for our evaluation[7].

3.2.1. Phonetics & Phonology

Studies outside Tamil, such as Begus et al. (2025), evaluated metalinguistic phonological abilities and showed that LLMs generalize over patterns like intervocalic gemination using synthetic words. PolyBench (Suvarna et al., 2024) further introduced tasks on grapheme-to-phoneme conversion and syllable counting, noting that phonological competence remains underexplored even for English. These approaches provide useful reference points for designing Tamil-specific phonological evaluations. However, Mmodel evaluations have yet to examine Tamil phonetics and phonology in detail, although some existing studies offer transferable insights. In SEA-HELM's LINDSEA suite reports that Tamil's script posed challenges for grapheme-level prompt formatting, adding further difficulty for phonology-related tasks.

3.2.2. Morphology

Tamil's morphological richness continues to pose difficulties for current LLMs. Results from the LINDSEA tests in SEA-HELM showed frequent errors in morphological agreement, including mismatches in case, gender and number (Leong et al., 2023; Susanto et al., 2025). Such problems appeared consistently across Question Answering (QA) and sentence completion tasks, indicating that the models lack a stable inflection mechanism for agglutinative languages.

PARIKSHA (Watts et al., 2024) reported a perfect acceptability score for GPT-4-Turbo on Tamil texts, but this self-assessment measure does not necessarily capture linguistic depth. The gap between surface acceptability and internal representation highlights the need for systematic probing of morphological understanding in Tamil.

3.2.3. Syntax

Tamil syntax, with its free word order and nested predicate structures, presents additional challenges for current models. The syntactic diagnostics in LINDSEA evaluate phenomena such as argument structure and filler–gap dependencies, areas where LLMs often underperform (Leong et al., 2023). Following the approach of BLiMP (Warstadt et al., 2020), these tests use minimal pairs to probe specific syntactic contrasts and expose weaknesses in handling long-distance dependencies or embedded clause scrambling.

Benchmarks like PAWS (Zhang et al., 2019) also inform Tamil syntax evaluation by showing that models tend to rely on surface word overlap rather than syntactic structure—a concern particularly relevant for Tamil, where flexible word order is grammatical. Begus et al. (2025) reported similar limitations, noting that LLMs struggle with recursion and structural ambiguity, consistent with the syntactic difficulties observed for Tamil.

[7] https://linguistics.ucla.edu/undergraduate/what-is-linguistics/

3.2.4. Semantics

Semantic understanding remains one of the most challenging aspects for Tamil, both in isolation and in inference-based settings. SEA-HELM evaluated Tamil through the IndicXNLI benchmark and reported that GPT-4o reached a score of 64.7, notably lower than its performance on Vietnamese and Indonesian (Susanto et al., 2025). The earlier BHASHA study observed even lower semantic comprehension (33.43%) for GPT-4 when English prompts were used (Leong et al., 2023). The model performed better, around 70%, on salient elements such as the emphatic particle *taan*, suggesting that high-salience, language-specific tokens are more reliably captured.

4. Data Creation

ILAKKANAM comprises 820 questions spanning Grades 1–13, each annotated by two trained linguists under five linguistic categories (see Table 1 for linguistic categories and their descriptions, and see Table 2 for examples under each category). Questions targeting factual knowledge are labeled as Facts to evaluate Tamil world knowledge in LLMs. This resource is designed to systematically assess both linguistic competence and culturally grounded knowledge in Tamil.

To build this evaluation resource, we developed a structured pipeline to convert school-level examination materials into a machine-readable dataset. The workflow involves three key stages: (1) collecting question papers from open educational archives, (2) digitizing and cleaning scanned documents and (3) structuring and filtering the finalized data for experimental use.

4.1. Data Collection

The dataset was built using school-level Tamil language examination questions from Grades 1–13 in Sri Lanka. The papers were sourced from the `Noolaham School`[8] section of `Noolaham.org`[9]. Two latest exam papers were selected per grade during the curation phase to ensure grade-wise coverage and topic diversity. In the digitization process, essay-type and structured questions were excluded, focusing instead on items that yield concrete, verifiable answers.

4.2. Digitization & Cleaning

The exam papers were available only as scanned PDFs because most were typed using non-Unicode Tamil fonts. To obtain machine-readable text, Google's Optical Character Recognition (OCR) engine was applied via Google Docs, chosen for its accessibility and ease of use. Each file was opened directly in Google Docs, which extracted text through its built-in OCR system.

Automated conversion introduced character, spacing, and formatting errors, which were manually corrected. A simple web interface was used to input cleaned questions and ensure the formatting consistency.

4.3. Data Structure & Filtering

To maintain consistency with the school examination paper format, the dataset incorporates nine distinct question types. This design also contributes to the broader task diversity beyond traditional Multiple-Choice Questions (MCQs), while still retaining a substantial proportion of the dataset. After automated filtration and manual correction, a final inspection was performed to remove both exact and near-duplicate questions.

Each datapoint was described using six key fields to ensure comprehensive information capture. A detailed description of these fields and their subfields is provided in Table 1, including the number of questions in each question type and linguistic category. Questions requiring lengthy written responses were adapted into multiple-choice format with non-obvious answer options to facilitate automated evaluation.

It should also be noted that questions related to Pragmatics, Discourse, Stylistics, and other higher-order aspects were incorporated into the Semantics category (L5; see Table 1 and Table 2). This decision was made because semantic interpretation is shaped not only by lexical and compositional meaning, but also by pragmatic inference, discourse context, and stylistic choices, all of which contribute to how meaning is ultimately constructed and understood. Consequently, this has led to an increased count of questions in the semantics category.

We also retained the original mark allocations from the papers as they could provide a direct measure of difficulty of questions.

After the extraction and validation phases were completed, the finalized dataset was exported in JSON format for experimental use.

5. Large Language Model Evaluation

This section outlines the setup, configuration, and methodology used to evaluate multiple LLMs on the curated Tamil question–answer dataset.

[8] https://noolaham.school/

[9] A digital archive of open Tamil educational resources.

Table 1: Dataset structure and categorical counts for the curated examination papers.

Field	Subfield	Description	Count
Paper ID	—	A composite identifier containing the school or institution name, grade level, and year of examination, enabling precise tracking of question origins.	—
Question Type ID	QT01	Fill in the blanks	13
	QT02	Provide answer based on the given set of letters/words	80
	QT03	Order the words/letters	24
	QT04	Question and Answer	182
	QT05	Sentence completion	3
	QT06	Rewrite with punctuation marks	13
	QT07	Multiple Choice Questions (MCQ)	469
	QT08	Question and Answer based on given paragraph	21
	QT09	True or False	15
Linguistic Category ID	L1	Phonetics	20
	L2	Phonology	32
	L3	Morphology	75
	L4	Syntax	169
	L5	Semantics	512
	F	Fact (questions testing knowledge of Tamil cultural, historical, or factual information)	12
Question Text	—	Question from the examination paper.	—
Answer	—	The ground truth or correct answer, validated by professional linguists.	—
Score	—	The point value assigned to each question in the original examination paper, preserved to maintain the weighted importance of different questions.	—

Table 2: Illustrative example for each linguistic category from the *ILAKKANAM* Dataset.

Category	Example Question from the Dataset
L1 (Phonetics)	பின்வருவனவற்றுள் வளைநா மூக்கொலியாக அமைவது எது? "Which of the following is a retroflex nasal?"
L2 (Phonology)	'க' கரம் வேறுபட்டு ஒலிக்கும் சொல் "A word in which the letter 'ka' sounds differently"
L3 (Morphology)	'கண்டான்' என்பதில் பகுதியாக அமைவது "lemma of the word *kaṇṭāṉ* (he) saw"
L4 (Syntax)	"நிமலன் நன்றாகப் படித்தான்.", "நிமலன் சிறந்த பெறுபேறு பெற்றான்." இவ்விரு வாக்கியங்களையும் இணைப்பதற்கு மிகப் பொருத்தமான இணைப்பிடைச் சொல் "'Nimalan studied well', 'Nimalan obtained excellent results' Which is the most appropriate conjunction to connect these two sentences?"
L5 (Semantics)	வேலைக்கு ஆட்சேர்ப்பது தொடர்பான அரசாங்கத்தின் உத்தியோகபூர்வ விளம்பரம் <இடைவெளி> வெளிவந்துள்ளது. "The official government advertisement regarding recruitment for the position has been published in the <blank>"
F (Fact)	புதிய ஆத்திசூடியின் ஆசிரியர், "The author of the new *Atticusudi*"

5.1. Evaluation Setup and Configuration

We utilized `Abacus.AI`[10] as a unified interface to access both open-source and closed-source models (See Appendix A). The list of evaluated models can be found in Table 3.

To guarantee consistency and factuality in responses, the generation temperature was fixed at 0. All other hyperparameters were left at their default values provided by the platform. To prevent information leakage, ground-truth answers were stored in a separate JSON file, locally, leaving only the questions accessible to the models. Each

[10] https://chatllm.abacus.ai

question file was fed systematically to every LLM under evaluation, and the generated responses were stored in separate files for later analysis. This setup allowed multiple models to be evaluated in parallel while preserving data integrity throughout the process.

Table 3: Model catalog grouped by Source Type and Provider.

Source Type	Provider	Model
Closed-Source	OpenAI	GPT-5
	Anthropic	Claude Sonnet 4.5
	Google	Gemini 2.5
	xAI	Grok 4
Open-Source	Meta	Llama 4
	DeepSeek	DeepSeek-V3
	Alibaba	Qwen 2.5 72B

Table 4: We report the overall results of each model through Score Percentage (SP). The numbers in parentheses indicate the actual count of correct answers out of 820 questions. Results of the best performing model is made bold.

Model	SP (/820)
Claude Sonnet 4.5	71.09 (579)
DeepSeek-V3	58.04 (491)
Gemini 2.5	**79.55 (659)**
Llama 4	60.67 (501)
OpenAI GPT5	75.94 (633)
Qwen 2.5	37.93 (320)
xAI Grok 4	78.15 (638)

5.2. Evaluation and Analysis

Model performance was evaluated by comparing each model's responses with the validated ground-truth answers (See Appendix B for system prompt). Scores were measured at several levels of detail to capture both overall and category-specific performance. In addition, we conducted a classification task where models were prompted to assign each question to one of six predefined categories (L1–L5 and F) in zero-shot settings (See Appendix C for system prompt).

Since not all questions from the original examination papers were included, the number of items differed across grades. To allow fair comparison, the grade-level scores were normalized to a 100-point scale before analysis. We used the following equation to obtain the Score Percentage for each analysis.

$$SP = \frac{S_o}{S_t} \times 100 \quad (1)$$

where S_o denotes the total score obtained by the model and S_t denotes the total attainable score.

6. Results and Discussion

We evaluate model performance across four complementary dimensions to obtain a comprehensive understanding of their linguistic and task-level behavior:

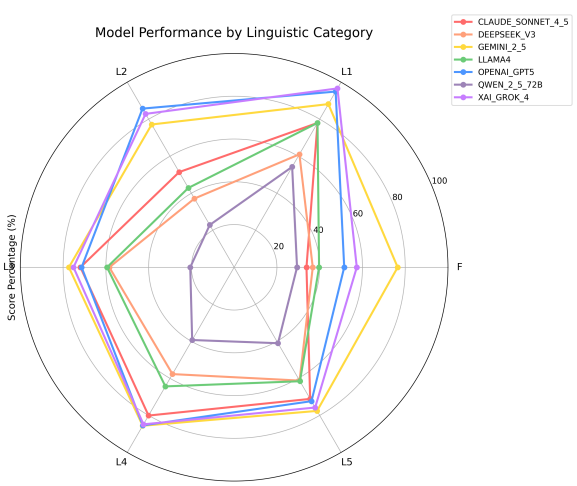
1. examine overall performance across the full dataset
2. analyze results by linguistic category (L1–L5 and F) to capture category-specific variations (refer Table 5 and Figure 1a)
3. report grade-wise performance to observe how models handle questions of varying complexity (refer Table 6 and Figure 1b)
4. present results from the linguistic category classification task, which assesses the models' ability to identify the underlying linguistic phenomenon in each question.

6.1. Overall Performance

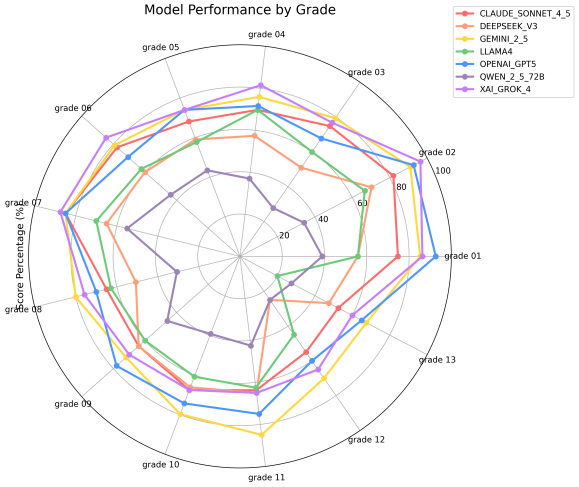
As presented in Table 4, the overall evaluation reveals clear variation in performance across models, with Gemini 2.5 achieving the highest score percentage of approximately 80%, reflecting superior linguistic understanding and factual precision-likely supported by Google's extensive multilingual and high-quality training data. Among all models, the closed-source group consistently occupies the top three ranks, highlighting their advantage in optimization, alignment, and dataset diversity. Among the open-source models, LLaMA 4 performed comparatively well with a score percentage of 60.67%, demonstrating strong generalization ability despite limited access to proprietary data. In contrast, Qwen 2.5, despite being a large-scale 72B model, recorded the lowest score (39.02%), reinforcing that model size alone does not guarantee better performance without effective linguistic grounding and diverse, representative training corpora.

6.2. Linguistic-wise Evaluation

Models were also evaluated across linguistic categories to assess how effectively they capture different aspects of linguistic understanding. The results are presented in Table 5 and Figure 1. The best-performing model overall, Gemini 2.5, demonstrated consistent performance across all categories, indicating a balanced grasp of Tamil language structure. In contrast, most other models showed weaker performance in the factual (F) category, which does not fall under linguistic analysis



(a) Score Percentage (SP) of each model across linguistic categories.



(b) Score Percentage (SP) of each model across grade levels.

Figure 1: Performance comparison of LLMs. (a) Category-wise results showing linguistic variation. (b) Grade-wise results showing variation across difficulty levels.

Table 5: Linguistic category-wise performance of each model through Score Percentage (SP). The numbers in parentheses indicate the actual count of correct answers. The best score in each linguistic category is made bold.

Model	**L1 (20)**	**L2 (32)**	**L3 (75)**	**L4 (169)**	**L5 (512)**	**F (12)**
Claude Sonnet 4.5	77.97 (15)	44.74 (15)	75.34 (53)	79.28 (129)	69.68 (362)	37.50 (5)
DeepSeek-V3	61.02 (13)	36.84 (13)	59.36 (42)	57.55 (99)	59.55 (319)	37.50 (5)
Gemini 2.5	88.14 (17)	71.05 (23)	**79.91 (57)**	**85.51 (143)**	**77.69 (410)**	**75.00 (9)**
Llama 4	77.97 (15)	28.95 (9)	62.10 (45)	63.78 (105)	61.12 (323)	32.50 (4)
OpenAI GPT5	94.92 (19)	78.95 (25)	75.80 (54)	84.51 (142)	72.62 (386)	57.50 (7)
Qwen 2.5 (72B)	54.24 (10)	31.58 (10)	21.92 (17)	40.44 (69)	39.22 (210)	35.00 (4)
xAI Grok 4	**96.61 (19)**	**84.21 (27)**	79.00 (56)	83.90 (136)	75.56 (393)	57.50 (7)
Average score	78.70	53.76	64.78	70.71	65.06	47.50

but assesses a model's Tamil world knowledge—including familiarity with poem authors, cultural references, literary works, and historical facts. When considering the linguistic dimensions alone, none of the models exceeded 80%, with Gemini 2.5 achieving the highest scores of 79.91% in L3 and 77.69% in L5, showing its relative strength in linguistic comprehension. The highest score within a linguistic category was observed in L1 (phonetics)(see Figure 1a), achieved by Grok 4 (96.61%), closely followed by GPT-5 (94.92%), reflecting their superior performance in this aspect.

In addition, as noted in previous studies, models continue to perform poorly on phonology and morphology tasks (see Appendix D for representative failure cases), likely due to the complex and rich morphological structure of Tamil. Although better performance is generally expected on semantic tasks—since models can leverage contextual information to infer meaning—the overall scores remain low. This may also be attributed to the fact that the semantic test set also contains pragmatic questions, which introduce additional challenges for Tamil.

6.3. Grade-wise Evaluation

Table 6 presents the grade-wise evaluation results of all LLMs, while Figure 1b illustrates the overall performance trend across grades. The grade-wise score percentage analysis shows that all models performed relatively well in Grades 1 and 2, which is expected since the questions at these levels are simpler and focus more on basic language skills that can be easily captured from training data. As the grade level increases, particularly from Grade 5 onwards, a noticeable decline in performance is observed across models. This corresponds to the increasing linguistic and conceptual complexity of questions that require a stronger command of Tamil grammar, vocabulary, and linguistic structure. The lowest scores appear around Grades

Table 6: We report the Score Percentage (SP) for each grade, with the number of correct answers shown in parentheses. The best result in each grade is highlighted in bold.

Grade	Claude Sonnet 4.5	DeepSeek-V3	Gemini 2.5	Llama 4	OpenAI GPT5	Qwen 2.5	xAI Grok 4
Gr1 (60)	74.74 (41)	55.79 (29)	85.26 (50)	55.79 (29)	**92.63 (55)**	38.95 (21)	86.32 (51)
Gr2 (58)	81.98 (46)	70.27 (41)	90.99 (53)	66.67 (39)	92.79 (54)	34.23 (20)	**96.40 (56)**
Gr3 (90)	74.84 (68)	50.97 (46)	**79.35 (70)**	60.00 (55)	67.74 (62)	27.74 (26)	76.77 (71)
Gr4 (120)	69.75 (91)	57.41 (76)	75.93 (98)	69.75 (89)	71.60 (94)	37.04 (48)	**81.48 (102)**
Gr5 (111)	68.18 (79)	59.09 (73)	**74.03 (89)**	57.79 (71)	**74.03 (88)**	43.51 (52)	**74.03 (88)**
Gr6 (37)	77.69 (29)	60.00 (22)	79.23 (29)	62.31 (23)	70.77 (26)	43.85 (16)	**84.62 (31)**
Gr7 (40)	85.00 (34)	65.00 (26)	85.00 (34)	70.00 (28)	85.00 (34)	55.00 (22)	**87.50 (35)**
Gr8 (40)	65.00 (26)	50.71 (21)	**80.00 (32)**	62.86 (25)	70.00 (28)	30.71 (13)	75.71 (30)
Gr9 (50)	64.00 (32)	64.00 (32)	72.00 (36)	60.00 (30)	**78.00 (39)**	46.00 (23)	70.00 (35)
Gr10 (74)	67.57 (50)	66.22 (49)	**79.73 (59)**	60.81 (45)	74.32 (55)	39.19 (29)	67.57 (50)
Gr11 (80)	63.75 (51)	65.00 (52)	**85.00 (68)**	62.50 (50)	75.00 (60)	42.50 (34)	65.00 (52)
Gr12 (20)	55.00 (11)	25.00 (5)	**70.00 (14)**	45.00 (9)	60.00 (12)	25.00 (5)	65.00 (13)
Gr13 (40)	52.50 (21)	47.50 (19)	**67.50 (27)**	20.00 (8)	65.00 (26)	27.50 (11)	60.00 (24)

5 and 13, which align with national-level examinations in Sri Lanka, where the questions are more challenging and require precise linguistic understanding. Even though Grade 11 (G.C.E. O/L) is also a national exam, it is less competitive, which is reflected in slightly better scores. We see that xAI Grok 4 performing best for lower grades 1-7, while Gemini 2.5 outperforms all he models at higher grades (8-13). Among all models, Gemini 2.5 maintained consistently high performance across grades, achieving above 67% even at the higher levels, reflecting its stronger adaptability to linguistic variation and question complexity compared to other models.

Table 7: Overall Accuracy Percentage (AP) for the linguistic classification task. Numbers in parentheses show correctly classified tags out of 820. Best-performing model is in bold.

Model	SP (/820)
Claude Sonnet 4.5	44.02 (361)
DeepSeek V3	27.93 (229)
Gemini 2.5	52.07 (427)
LLaMA 4	51.59 (423)
OpenAI GPT-5	**65.61 (538)**
Qwen 2.5 72B	52.07 (427)
xAI Grok 4	61.59 (505)

6.4. Linguistic Category Classification

In addition to the primary evaluation task, an additional experiment was conducted to better understand each model's ability to capture linguistic awareness (refer Table 7). In this setup, the models were instructed to assign an appropriate linguistic tag (from L1 to L5) to each question, with any item not fitting a linguistic category explicitly directed to be tagged as F. A notable observation emerged from this experiment: while Gemini 2.5 performed exceptionally well in answering questions during the main evaluation, it ranked second in this linguistic categorization task, falling behind GPT-5, with 111 fewer correctly tagged questions. This finding reinforces the argument that Gemini 2.5's strong performance may not stem from genuine linguistic understanding, but rather from its extensive training data coverage (Rane et al., 2024). Since the evaluation questions are based on Tamil school examination papers, they are likely to follow predictable patterns and exhibit limited novelty, making them easier for Gemini to match with seen data. In contrast, the linguistic tagging task requires deeper analytical ability and true understanding of linguistic structures—skills that cannot be derived from memorized or surface-level data. In this respect, GPT-5 demonstrated stronger linguistic awareness and interpretive precision.

7. Copyrights

All the questions used in this work have been sourced from publicly available materials that are licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

8. Conclusion

We introduce *ILAKKANAM*, a Tamil linguistic benchmark dataset consisting of 820 manually curated questions from Sri Lankan school-level Tamil grade-wise examination papers. We perform an extensive quantitative evaluation on both closed-source and open-source LLMs and show that there is a clear gap between them in terms of Tamil linguistic performance. We analyze the overall results, discuss model performance across linguistic categories and grade levels, and present observations from the linguistic categorization task. Our studies show that Gemini 2.5 performs the best on our benchmark dataset, and OpenAI GPT-5 performs the best on the linguistic category classification. We find no clear relationship between a model's performance on the linguistic tasks and it's ability to classify these questions into their respective linguistic categories, further suggesting that LLM performance may not stem from linguistic understanding but rather from broad exposure to training data. We hope that both the *ILAKKANAM* dataset and our analyses help researchers better understand the limitations of current models and encourage further efforts toward evaluating and benchmarking LLMs for the Tamil language. Our implementation and dataset are accessible at: https://github.com/LTG-UoJ/ILAKKANAM-Public, where the dataset includes only extracted examination questions to minimise the risk of data leakage.

9. Limitations

First, two recent papers were selected from each grade level, which may not fully capture the breadth and diversity of Tamil linguistic phenomena. As a result, certain aspects of language use and structure may not be adequately represented in the current dataset. We are actively expanding the question bank to improve linguistic coverage and representation.

Second, our analysis focused on five core linguistic dimensions: phonetics, phonology, morphology, syntax, and semantics. Extended linguistic areas, such as pragmatics and stylistics, were grouped under semantics because of their limited presence in the dataset. This grouping may reduce the granularity of analysis for these higher-level aspects.

Third, the performance scores reported in this study may change over time as models continue to improve through updates and new releases. Therefore, the results should be interpreted as a snapshot of model capabilities at the time of evaluation. In future work, we plan to develop a continuously updated evaluation dashboard to track model performance over time.

10. Future Work

To address the current limitations, future work will prioritise expanding the dataset by incorporating a larger number of examination papers across multiple years and grade levels. This will improve the diversity and representativeness of Tamil linguistic phenomena within the question bank, enabling more comprehensive evaluation of LLMs across linguistic dimensions.

In addition to the existing quantitative evaluation, future research will incorporate qualitative error analysis to provide deeper insights into model behaviour and identify systematic linguistic weaknesses. This will support a more fine-grained assessment of model performance across different linguistic dimensions.

Furthermore, future work will also extend the evaluation by including small and mid-sized open-source language models. Assessing their performance is essential for low-resource settings, where computational and infrastructural constraints necessitate efficient yet capable models.

11. Acknowledgements

This research is part of the Sri Lankan Tamil Corpus project[11] at the University of Jaffna and is supported by a Google Research Scholar Award to Kengatharaiyer Sarveswaran. We thank Noolaham Foundation for making digitized school examination papers publicly available. We also extend our sincere gratitude to Ms. Ramanan Sivaranjani for her assistance in obtaining the missing examination papers.

12. Bibliographical References

References

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024.

[11] https://sites.google.com/univ.jfn.ac.lk/talkcorpus/home

The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

Gasper Begus, Maksymilian Dabkowski, and Ryan Rhodes. 2025. Large linguistic models: Investigating llms' metalinguistic abilities. *IEEE Transactions on Artificial Intelligence*, page 1–15.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models.

Claude. 2025. Introducing Claude Sonnet 4.5 — anthropic.com. https://www.anthropic.com/news/claude-sonnet-4-5. Accessed 2025-10-03.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, ..., and Wesley Helmholz. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, ..., and Zizheng Pan. 2025. Deepseek-v3 technical report.

Xiyan Fu and Wei Liu. 2025. How Reliable is Multilingual LLM-as-a-Judge?

Omid Ghahroodi, Marzia Nouri, Mohammad V. Sanian, Alireza Sahebi, Doratossadat Dastgheib, Ehsaneddin Asgari, Mahdieh Soleymani Baghshah, and Mohammad Hossein Rohban. 2024. Khayyam Challenge (PersianMMLU): Is Your LLM Truly Wise to The Persian Language? *ArXiv*, abs/2404.06644.

Grok4. 2025. Grok 4. https://x.ai/news/grok-4. Accessed 2025-10-05.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A Survey on LLM-as-a-Judge.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Meng Ji, Meng Ji, Pierrette Bouillon, and Mark Seligman. 2023. *Cultural and Linguistic Bias of Neural Machine Translation Technology*, Studies in Natural Language Processing, page 100–128. Cambridge University Press.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Fajri Koto, Nurul Aisyah, Haonan Li, and Timothy Baldwin. 2023. Large language models only pass primary school exams in Indonesia: A comprehensive test on IndoMMLU. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12359–12374, Singapore. Association for Computational Linguistics.

Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. ArabicMMLU: Assessing massive multitask language understanding in Arabic. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5622–5640, Bangkok, Thailand. Association for Computational Linguistics.

Wei Qi Leong, Jian Gang Ngui, Yosephine Susanto, Hamsawardhini Rengarajan, Kengatharaiyer Sarveswaran, and William Chandra Tjhi. 2023. BHASA: A Holistic Southeast Asian Linguistic and Cultural Evaluation Suite for Large Language Models.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, Bangkok, Thailand. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hud-

son, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

IM Kamala Liyanage. 2014. Education system of sri lanka: strengths and weaknesses. *Educ Syst Sri Lanka*, pages 116–40.

Llama4Herd. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/. Accessed 2025-10-01.

OpenAI. 2022. ChatGPT: Optimizing Language Models for Dialogue. Accessed: 2024-08-15.

OpenAI. 2025. OpenAI Introducing GPT5. https://openai.com/index/introducing-gpt-5/. Accessed 2025-10-22.

Ashmari Pramodya, Nirasha Nelki, Heshan Shalinda, Chamila Liyanage, Yusuke Sakai, Randil Pushpananda, Ruvan Weerasinghe, Hidetaka Kamigaito, and Taro Watanabe. 2025. SinhalaMMLU: A Comprehensive Benchmark for Evaluating Multitask Language Understanding in Sinhala.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Nitin Rane, Saurabh Choudhary, and Jayesh Rane. 2024. Gemini versus chatgpt: applications, performance, architecture, capabilities, and implementation. *Journal of Applied Artificial Intelligence*, 5:69–93.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. 2025. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2025. KMMLU: Measuring massive multitask language understanding in Korean. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104, Albuquerque, New Mexico. Association for Computational Linguistics.

Yosephine Susanto, Adithya Venkatadri Hulagadri, Jann Railey Montalan, Jian Gang Ngui, Xianbin Yong, Wei Qi Leong, Hamsawardhini Rengarajan, Peerat Limkonchotiwat, Yifan Mai, and William Chandra Tjhi. 2025. SEA-HELM: Southeast Asian holistic evaluation of language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12308–12336, Vienna, Austria. Association for Computational Linguistics.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. PhonologyBench: Evaluating phonological skills of large language models. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop*

BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Ishaan Watts, Varun Gumma, Aditya Yadavalli, Vivek Seshadri, Manohar Swaminathan, and Sunayana Sitaram. 2024. PARIKSHA: A large-scale investigation of human-LLM evaluator agreement on multilingual and multi-cultural data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7900–7932, Miami, Florida, USA. Association for Computational Linguistics.

Arda Yüksel, Abdullatif Köksal, Lütfi Kerem Senel, Anna Korhonen, and Hinrich Schuetze. 2024. TurkishMMLU: Measuring massive multitask language understanding in Turkish. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7035–7055, Miami, Florida, USA. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

13. Appendices, Software, and Data

A. Prompting Interface

We used the Abacus.AI API to query multiple models through a unified interface. A simplified version of our querying function is shown below:

```
client = ApiClient(api_key)

def query_model(client, model_name, question, system_prompt, temperature):
    response = client.evaluate_prompt(
        prompt=question,
        system_message=system_prompt,
        llm_name=model_name,
        temperature=temperature
    )
    return response.content
```

B. System Prompt 1

The following system prompt was used for the task, where models responded to exam-style question papers.

```
system_prompt = (
    "You are a helpful and concise assistant."
    "Provide only the answer - do not include explanations or full sentences."
    "For generation tasks, give a brief, direct response."
    "For multiple-choice questions (MCQs), return only the option number or letter
       corresponding to the correct answer."
)
```

C. System Prompt 2

The following system prompt was used for the task, where models performed zero-shot classification into predefined categories (L1-L5 and F, see Table 1).

```
system_prompt = (
    "You are a precise linguistic classifier. "
    "Do not answer the question. Your only task is to analyze the given question
       text "
    "and assign it to one linguistic category based on its content. "
    "Return only the category ID as follows:\n"
    "L1: Phonetics\n"
    "L2: Phonology\n"
    "L3: Morphology\n"
    "L4: Syntax\n"
    "L5: Semantics\n"
    "F: Factual or any other non-linguistic content.\n"
    "Respond with only the ID (e.g., 'L3'), without explanations, reasoning, or
       extra text."
)
```

D. Phonology (L2) and Morphology (L3) Representative Failure Cases

Table 8: Examples of phonology & morphology questions where all models produced incorrect answers.

Question	Correct Answer	Model Answers
பின்வரும் வினாவிற்கான மிகச்சரியான விடையின் இலக்கத்தை மட்டும் தருக. 'க' என்ற எழுத்து உச்சரிப்பில் வேறுபட்டு ஒலிக்கும் சொல், "Select the number corresponding to the most appropriate answer. Which word contains the letter 'ka' pronounced differently?" 1. பங்**க**ம் (*paŋgam*) 2. பே**க**ன் (*pēxɛn*) 3. தா**க**ம் (*tāxam*) 4. அ**க**ம் (*axam*)	1	Claude: 2 DeepSeek: 4 Gemini: 2 LLaMA: 2 GPT-5: 2 Qwen: 3 Grok: 2
பின்வரும் வினாவிற்கான மிகச்சரியான விடையின் இலக்கத்தை மட்டும் தருக. பின்வருவனவற்றுள் ஐகாரக்குறுக்கம் இடம்பெறாத சொல், "Which of the following words does not contain the diphthong 'ai' contraction?" 1. ஐப்பசி (*aippasi*) 2. வைகாசி (*vaixāsi*) 3. மாயை (*māyai*) 4. பை (*pai*) 5. வசை (*vasai*)	1	Claude: 5 DeepSeek: 3 Gemini: 4 LLaMA: 2 GPT-5: 3 Qwen: 3 Grok: 3
பிரித்தெழுதுக. "Split the compound word into its parts." வடகிழக்கு Northeast	வடக்கு + கிழக்கு	Claude: வடம் + கிழக்கு DeepSeek: வட + கிழக்கு Gemini: வடக்கின் + கிழக்கு LLaMA: வடகு + இழக்கு GPT-5: வட + கிழக்கு Qwen: வடகி ழக்கு Grok: வட + கிழக்கு
பின்வரும் வினாவிற்கான மிகச்சரியான விடையின் இலக்கத்தை மட்டும் தருக. திரிதல் விகாரம் இடம் பெறாத சொல், "Which word does not undergo phonological alternation?" 1. கடற்கரை "Seashore" 2. மரங்கள் "Trees" 3. கொண்டேன் "Held (I)" 4. இருமல் "Cough" 5. ஓடிப்போ "Run-Go"	5	Claude: 4 DeepSeek: 3 Gemini: 4 LLaMA: 4 GPT-5: 4 Qwen: 3 Grok: 4

A Feature-Fusion Ensemble Approach for Tamil Hate Speech Detection

Sathasivam Nerujan and Kengatharaiyer Sarveswaran
University of Jaffna
sathasivamnerjan35@gmail.com and sarves@univ.jfn.ac.lk

Abstract

Detecting online toxicity in morphologically rich, low-resource languages like Tamil remains a major computational challenge. Standard transformer models often struggle with sub-word fragmentation, which can dilute the semantic intensity of regional insults and out-of-vocabulary slang. To mitigate this limitation, we train a multi-layer hybrid framework that fuses the deep contextual representations of L3Cube-TamilBERT with the character-level robustness of FastText embeddings. Our architecture leverages Last-4 Layers averaging and a dual pooling strategy (Mean + Max) to capture both global sentence intent and extract high-activation spikes of offensive cues typically lost in single layer representations. Experiments show that this hybrid model achieves a Macro-F1 of 0.7883, notably enhancing Hate Recall (0.7503) for detection of offensive content. Additionally, as reported by other studies, stacking ensemble achieves peak hate precision (0.9296), providing a high accuracy alternative for moderation scenarios requiring minimal false positives. By combining deep contextual hidden states with FastText embeddings, the proposed feature-fusion ensemble approach with multi-layer hybrid framework approach establishes a new benchmark for hate speech detection for Tamil.

Keywords: Tamil Hate Speech, Transformer Fusion, FastText, Dual Pooling

1. Introduction

Hate speech on social media causes direct harm to targeted communities and can amplify offline discrimination, making timely automated detection a critical societal and platform-level need. Surveys and empirical studies show that automatic hate speech detection is feasible, but remains error-prone and highly sensitive to domain shifts, annotation policies, and evolving language. Moreover, building reliable systems is a data intensive process that depends on large, carefully annotated corpora to capture nuanced, context dependent judgments and emerging forms of abusive language (Fortuna and Nunes, 2018).

Hate speech detection in Tamil remains challenging, apart from its low-resource nature, due to its agglutinative morphology, rich inflectional structure, and highly informal digital usage characterized by Tamil-English code-mixing, slang, and creative spellings (Maria Nancy et al., 2025). These properties create substantial vocabulary mismatch, where conventional transformer models often struggle (Xu, 2026) to capture culturally specific offensive expressions and morphologically complex word forms. In particular, sub-word tokenization can fragment region specific insults, weakening their semantic intensity and reducing detection sensitivity.

While pretrained transformer models provide strong contextual representations, relying solely on the final layer's `[CLS]` token is often insufficient for capturing the full semantic nuance of complex sentences. As noted by (Reimers and Gurevych, 2019), the default `[CLS]` output in BERT-based architectures frequently yields poor sentence-level embeddings, often underperforming compared to simple mean pooling. This limitation is particularly pronounced in Tamil hate speech detection, where localized offensive cues can be diluted within longer, agglutinative sequences or lost due to out-of-vocabulary slang. Consequently, recent work in toxic language moderation has moved toward hybrid feature-based approaches. For instance, (Iftikhar et al., 2025) demonstrates that fusing transformer ensembles with hybrid feature sets is essential for navigating the volatility of digital discourse. These theoretical and empirical gaps motivate our use of a more robust representation strategy, combining last-4 layers averaging with dual pooling and sub-word embeddings.

To address this, we propose a hybrid framework that integrates transformer-based contextual embeddings with FastText sub-word representations. The architecture incorporates last-4 layers averaging to enhance representational stability and a dual pooling strategy (mean and max) to capture both global sentence semantics and localized intensity peaks. Experimental results show that the proposed hybrid design achieves a Macro-F1 score of 0.7883, which is competitive with more complex ensemble-based approaches, while maintaining a moderate balance between hate recall (0.7503) and precision (0.6375). Unlike stacking-based models that strongly favor precision or large multilingual ensembles that prioritize recall at higher computational cost, our approach provides stable performance using a single

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 190–197
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

language-specific transformer, making it suitable for practical Tamil hate speech detection settings. Unlike prior work, our approach combines multi-layer transformer features, dual pooling, and FastText embeddings in a single framework, enabling better handling of semantic and morphological aspects of Tamil code-mixed text.

The primary contributions of this paper are: (1) We introduce a hybrid feature fusion architecture that combines transformer-based contextual embeddings with FastText sub-word representations to better handle Tamil's agglutinative morphology and informal, code-mixed social media text; (2) We propose a flexible multi layer representation strategy based on last-4 layers averaging and dual pooling (mean and max), and evaluate its performance alongside alternative layer averaging and pooling approaches; and (3) We present a systematic comparative analysis of single, hybrid, and ensemble modeling strategies, highlighting their precision, recall trade-off for Tamil hate speech detection.

2. Related Work

Recent research in Tamil hate speech detection has shifted from traditional machine learning algorithms, such as SVM and Random Forest, to transformer-based architectures to better handle code-mixed and morphologically rich social media text. Sangeetham et al. (Sangeetham et al., 2024) demonstrated this transition using the HASOC 2021 dataset (approximately 5.8K comments), where their transformer-based ensemble achieved an F1-score of 0.68, showing that conventional models struggle to capture the contextual nuances of Tamil digital discourse.

To address these limitations, multilingual transformer models such as mBERT and XLM-RoBERTa have been widely adopted. Karim et al. (Karim et al., 2025) leveraged ensembles of these models on the LT-EDI 2025 dataset (over 10K samples focused on caste and migration related hate speech), achieving a competitive F1-score of 0.803 through majority voting. In parallel, language-specific models such as L3Cube-TamilBERT have shown improved linguistic alignment; Roy et al. (Roy et al., 2025) reported an F1-score of 0.79 on a Tamil offensive language benchmark, outperforming multilingual baselines.

Despite these advances, handling out-of-vocabulary slang in Tamil-English code-mixed text remains challenging. Iftikhar et al. (Iftikhar et al., 2025) explored hybrid architectures on an 8K sample social media corpus, combining transformer contextual embeddings with FastText sub-word representations, achieving an F1-score of 0.77.

More recently, S et al. (S et al., 2025) proposed a multi-scale hybrid approach, fusing TF-IDF features at multiple scales with embeddings from five transformer models. Their three-level hierarchical ensemble integrates classical and contextual features across classifiers, achieving a peak F1-score of 0.818 on the LT-EDI 2025 shared task. The dataset used in this study is not publicly available, highlighting the challenges of benchmarking low-resource, code-mixed Tamil-English hate speech detection.

3. Approach

3.1. Dataset

We use the Tamil Offensive Speech Detection dataset, which consists of Tamil YouTube comments in native script, Romanized Tamil, and Tamil-English code-mixed text, annotated with binary labels (0 = non-offensive, 1 = offensive). The dataset is compiled from Tamil Wikipedia articles and shared-task resources, including the Dravidian-CodeMix HASOC @ FIRE 2020 shared task (Mandalam et al., 2020) and the Dravidian-CodeMix sentiment analysis task at FIRE 2020 (Chakravarthi et al., 2021). The full dataset is publicly available through Kaggle[1].

The original training split contains 6,649 offensive (label 1) and 21,226 non-offensive (label 0) instances. For evaluation, we utilize the official file `tamil_offensive_speech_val.csv` provided in the release (Hata, 2020), which comprises 1,684 offensive and 5,285 non-offensive instances. Table 1 summarizes the final distribution of these records after our preprocessing and duplicate removal steps. In addition to the Kaggle dataset, we also evaluate our model on the HASOC Code-Mixed Tamil dataset to ensure fair comparison with prior work, as several existing approaches are benchmarked on this dataset.

3.2. Preprocessing

We remove duplicate entries from both the training and validation splits, as the provided validation file also contains duplicates. Although it is common practice to remove emojis during preprocessing, we retain them because they convey affective cues that may be informative for offensive speech detection, as discussed in (Amalia et al., 2025).

3.3. Modeling Strategies

To systematically analyze the effectiveness of different representation learning paradigms for Tamil

[1] https://www.kaggle.com/datasets/eshikanahata/tamil-offensive-speech-detection

Table 1: Dataset label counts after preprocessing.

Split	Hate (1)	Non-hate (0)
Training	5953	18942
Validation	662	2105
Testing	1678	5181

hate speech detection, we experiment with multiple modeling strategies. These strategies are designed to evaluate the impact of fine-tuning, feature extraction, ensemble learning, and hybrid representation fusion.

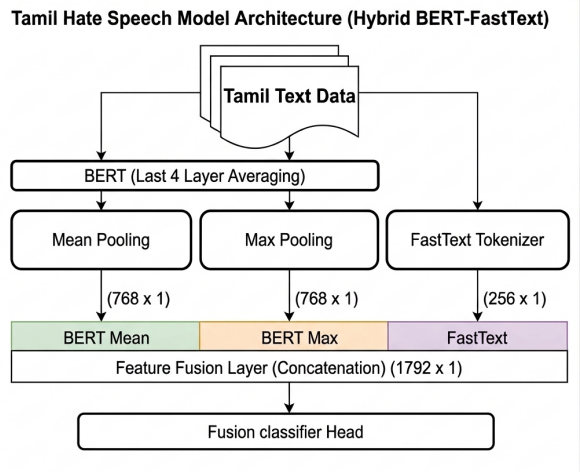


Figure 1: Overview of the proposed hybrid architecture.

3.3.1. Transformer Fine-Tuning

We fine-tune pretrained transformer models, including TamilBERT, MuRIL, and XLM-R, using a task-specific classification head. Each model is trained end-to-end with a softmax output layer for binary classification (Hate vs. Non-Hate), serving as strong neural baselines for comparison.

To combine the unique strengths of different architectures, we also developed a stacking-based ensemble. This is achieved by concatenating the sentence-level embeddings extracted from MuRIL and XLM-R. These fused representations are then processed by a Multi-Layer Perceptron (MLP) meta-classifier. By utilizing nonlinear hidden layers, the MLP learns complex interactions between the stacked features. This feature-level fusion approach yielded our peak Hate Precision of 0.9296, proving highly effective at balancing the specialized linguistic insights of both backbone models.

3.3.2. Hybrid Transformer FastText Models

To address the limitations of transformer tokenization on informal and out-of-vocabulary Tamil slang, we explore hybrid architectures that integrate transformer-based contextual embeddings with FastText sub-word representations. FastText leverages character-level n-grams to model subword patterns, enabling the model to capture both semantic context and morphological variations present in code-mixed Tamil text.

3.3.3. Proposed Feature-Level Fusion Architecture

Building on the hybrid framework, our approach incorporates last-4 layers averaging and dual pooling over transformer hidden states, followed by feature fusion with FastText embeddings. This fused representation is passed to an MLP classifier, enabling effective interaction between contextual and sub-word features for hate speech detection.

3.3.4. Transformer Backbone and Layer Averaging

We employ L3Cube-TamilBERT as the primary transformer encoder. To capture richer syntactic and semantic information, we extract hidden states from the last-4 transformer layers and compute their average.

Let H_i denote the hidden representation of the i^{th} transformer layer. The layer averaged representation is computed as:

$$H_{\text{avg}} = \frac{1}{4} \sum_{i=L-3}^{L} H_i$$

This multi-layer averaging strategy provides a more contextual representation, which is particularly beneficial for morphologically rich languages such as Tamil.

3.3.5. Dual Pooling Mechanism

Given the averaged hidden representation $H_{\text{avg}} \in \mathbb{R}^{T \times D}$, where T denotes the sequence length and D the embedding dimension, we apply a dual pooling strategy to obtain a fixed-length sentence representation.

The mean pooled representation is computed as:

$$\mu_{\text{pool}} = \frac{1}{T} \sum_{t=1}^{T} h_t$$

The max pooled representation is computed as:

$$M_{\text{pool}} = \max_{t=1}^{T} (h_t)$$

The final transformer-based sentence representation is obtained by concatenation:

$$V_{\text{BERT}} = [\mu_{\text{pool}}; M_{\text{pool}}]$$

Table 2: Performance comparison of evaluated models

Strategy Category	Model Architecture	Macro F1	Hate Recall	Hate Precision
Hybrid (Proposed)	TamilBERT + FastText (Layer Avg)	0.7883	0.7503	0.6375
Hybrid (Ensemble)	Hate-BERT + FastText (Layer Avg)	0.7888	0.7586	0.6346
Single Transformer	L3Cube-TamilBERT	0.7880	0.7384	0.6430
Single Transformer	mBERT	0.7639	0.6434	0.6432
Single Transformer	MuRIL	0.7770	0.6895	0.6456
Hybrid	MuRIL + FastText	0.7798	0.7682	0.6118
Meta-Classifier	MuRIL + XLM-R Stacking	0.7650	0.8008	0.9296
TF-IDF Fusion	MuRIL + XLM-R + TF-IDF	0.7004	0.5662	0.5373
Hybrid (Multilingual)	MuRIL + XLM-R + FastText	0.7867	0.8948	0.8960

This dual pooling strategy captures both the overall semantic context and localized high activation signals corresponding to offensive or toxic expressions.

3.3.6. FastText Sub-word Branch

The FastText branch generates sentence embeddings using character-level n-grams, capturing sub-word information for morphologically rich and out-of-vocabulary tokens. For each input sentence, a 300-dimensional embedding $V_{\text{FT}} \in \mathbb{R}^{300}$ is obtained.

To align with transformer-based features, V_{FT} is projected into a lower dimensional, task-specific space using a feed-forward layer with batch normalization, ReLU activation, and dropout:

$$H_{\text{FT}} = \text{Dropout}(\text{ReLU}(\text{BatchNorm}(WV_{\text{FT}} + b)))$$

where $W \in \mathbb{R}^{256\times300}$ and $b \in \mathbb{R}^{256}$. The resulting representation $H_{\text{FT}} \in \mathbb{R}^{256}$ is passed directly to the fusion stage.

3.3.7. Feature Fusion and Classification

The final fused feature representation is constructed by concatenating the dual pooled transformer outputs and the processed FastText embeddings:

$$V_{\text{fused}} = [V_{\text{BERT}} \parallel H_{\text{FT}}]$$

where $V_{\text{BERT}} \in \mathbb{R}^{1536}$ represent the mean and max pooled hidden states from the last four transformer layers, respectively, yielding a combined vector $V_{\text{fused}} \in \mathbb{R}^{1792}$.

This fused vector is passed to a Multi-Layer Perceptron (MLP) classifier with a hidden layer of 512 units, ReLU activation, and dropout. The final linear layer maps the 512-dimensional state to 2 output logits representing the classes (Hate vs. Non-Hate). During training, to directly combat dataset class imbalance, the model is optimized using a weighted cross-entropy loss function.

4. Results and Discussion

4.1. Performance Overview

Table 2 presents a comprehensive comparison of all evaluated architectures trained on the datasets detailed in Table 1. Given the class imbalance, the Macro F_1-score serves as our primary evaluation metric. Although the ensemble utilizing HateBERT[2] yielded the highest Macro F_1 of 0.7888, we did not select it as our best model. This backbone was pre-trained on YouTube and Twitter hate speech corpora, introducing a potential risk of data leakage or overlap with our test set. Therefore, our proposed `TamilBERT + FastText` framework (Macro F_1: 0.7883) stands as the most reliable architecture.

4.2. Comparative Analysis

4.2.1. Impact of Language-Specific Pretraining

Single L3Cube-TamilBERT achieved a Macro F1-score of 0.7880, outperforming the multilingual baseline mBERT (0.7639). This performance gap highlights the importance of language-specific pretraining, as Tamil only corpora allow the model to better capture agglutinative morphology, character level patterns, and syntactic constructions that are diluted in multilingual vocabularies.

4.2.2. Ablation Study and Architectural Analysis

To analyze the contribution of each architectural component, we evaluated multiple configurations using TamilBERT as a fixed backbone (Table 3). This systematic decomposition reveals how hybrid feature fusion and specific pooling strategies impact the model's ability to identify offensive content in complex linguistic environments.

The results indicate that the integration of FastText embeddings provides the most significant

[2] https://huggingface.co/mdosama39/tamil-bert-Caste-HateSpech_LTEDi-tamil

Table 3: Ablation study of architectural components using TamilBERT backbone

Model Strategy	Macro-F1	Prec.(Hate)	Rec.(Hate)
BERT + FastText	**0.7864**	0.6392	0.7390
BERT + Last-4 Layers Mean	0.7836	0.6621	0.6889
BERT + Dual Pooling	0.7755	0.5866	0.8153
BERT (Standard)	0.7647	0.5640	0.8302
BERT + Last Layer Mean	0.7356	0.5205	0.8254

overall performance gain. This effectiveness is largely attributed to its ability to mitigate out-of-vocabulary (OOV) slang and creative spellings in Tamil social media text. While standard transformer tokenizers often fragment such words into non-semantic sub-word units, FastText preserves essential morphological patterns through character n-grams.

Furthermore, leveraging the Last-4 Layers Mean enhances representation stability. By moving beyond the final hidden state which can be overly specialized to the pretraining objective the model captures deeper contextual signals. While dual pooling improves recall by emphasizing localized high-activation features, it introduces a minor reduction in precision due to increased sensitivity to noisy signals. Overall, the fusion of FastText with multi-layer contextual embeddings creates the most balanced representation for this morphologically rich and code-mixed language.

4.2.3. Dual Pooling and Multi-Layer Representation

Table 4: Effect of different pooling strategies using TamilBERT backbone

Pooling Method	Macro-F1
CLS Token	0.7647
Mean Pooling	0.7356
Dual Pooling (Mean + Max)	0.7755

Table 5: Effect of layer combination strategies using TamilBERT backbone

Layer Strategy	Macro-F1
Last Layer	0.7356
Last-4 Layers	0.7836

To isolate the contribution of each component, we evaluated pooling strategies and layer combination methods independently using TamilBERT as a fixed backbone. As shown in Table 4, dual pooling outperforms both CLS token and mean pooling, demonstrating its ability to capture both global sentence semantics and localized high activation features. This suggests that relying solely on the `[CLS]` representation is insufficient for modeling sentence-level meaning, consistent with prior findings (Reimers and Gurevych, 2019).

Similarly, Table 5 shows that combining the last-4 transformer layers significantly improves performance over using only the final layer, highlighting the importance of leveraging richer contextual representations. We do not extend the aggregation beyond the last-4 layers, as prior work (Tenney et al., 2019) has shown that higher transformer layers capture more task-relevant semantic information, while incorporating lower layers provides diminishing returns due to redundancy in lower-level features. This observation is consistent with existing findings and supports our design choice of restricting layer fusion to the last-4 layers.

Although dual pooling and multi-layer fusion provide complementary benefits, their improvements are not strictly additive indicating partial redundancy between learned features, suggesting partial feature overlap. Overall, these results emphasize that careful selection of pooling and layer strategies is critical for robust representation learning.

4.2.4. Qualitative Analysis

Table 6: Qualitative examples where the hybrid model correctly predicts non-hate cases

Text	Label
dhanush looks like gramathu vayasana thathaa	0
தமிழனை தமிழனை வைத்து அழிங்கிர சூழ்ச்சிதான் இது.....சாதி.....பா.ரஞ்சித் எதிர் மோகன்.....ஒரு நாள் எல்லா தமிழ் சமுகமும் கண்ணிர் விடும்....தன்னை தானே அழித்துகொண்டோம் என்று....	0
vvedalam teaser likes 145k but bairavaa teaser likes 171k beated vedalam teaser in just 2 days!!!!!	0

As shown in Table 6, the proposed hybrid model correctly classifies challenging non-hate examples that are misclassified by other architectures evaluated in Table 3. These errors typically arise due to informal language, cultural references, or noisy social media patterns.

Transformer only models tend to over predict offensive content due to fragmented tokenization, while FastText only or partial fusion models fail to capture deeper contextual cues. In contrast, the hybrid architecture effectively combines sub-word information with multi-layer contextual representations, allowing it to handle such nuances more accurately.

These examples further validate that the full hybrid configuration is more robust than its individual components, as reflected in the ablation results.

Table 7: Comparison on HASOC Code-Mixed Tamil dataset (SubTask 2)

Model / Study	Methodology	Macro F1
Our Hybrid Model	TamilBERT + FastText + L4 Fusion	0.8955
Siva@HASOC	XLM-R + mBERT (Ensemble + Transliteration)	0.8800
MUCIC@HASOC	Logistic Regression / BERT	0.6790
PSG@HASOC	MuRIL + Mean Pooling	0.6700

4.2.5. Sensitivity to Code-Mixed vs Pure Tamil Inputs

Table 8: Performance comparison across Pure Tamil and Code-Mixed inputs

Input	Macro-F1	Prec.(Hate)	Rec.(Hate)	Accuracy
Pure Tamil	0.7790	0.5301	0.7719	0.8813
Code-Mixed	0.7809	0.6172	0.7730	0.8216

Table 8 shows the model's performance across Pure Tamil and Code-Mixed inputs. Despite differences in data distribution, the model achieves nearly identical Macro-F1 scores, indicating low sensitivity to input type. While precision is higher for code-mixed text, recall remains consistent across both settings. Overall, this demonstrates that the model generalizes effectively to both pure Tamil and code-mixed social media content.

4.2.6. Fair Comparison with Existing Models

To ensure a fair comparison with existing models on Tamil hate speech detection, we first evaluated a publicly available pretrained model, Hate-speech-CNERG/deoffxlmr-mono-tamil (Saha et al., 2021). Although this model reports a Weighted F1-score of 0.76 (and 0.78 with ensembling), we observed an high Macro F1-score of 0.98 when evaluated on our test set, indicating that the model was likely trained on overlapping data.

To avoid such data leakage and ensure a consistent evaluation setting, we instead adopt the HASOC Code-Mixed Tamil dataset (SubTask 2), which is widely used in prior work. We then apply our hybrid methodology on the same dataset and compare it with previously reported results.

As shown in Table 7, our hybrid model achieves the highest Macro-F1 score (0.8955) with TamilBERT, outperforming prior approaches on the same dataset. While Siva@HASOC (Sai and Sharma, 2020) reports competitive results using multilingual ensembles with transliteration, details of their Exact test split are not publicly available, limiting reproducibility.

Compared to earlier systems such as MUCIC (Balouchzahi et al., 2022) and PSG@HASOC ((Benhur and Sivanraju, 2021)), our approach shows clear improvements by integrating subword representations and multi-layer contextual features, enabling more robust handling of code-mixed and morphologically complex Tamil text. Unlike prior work, our method does not rely on external transliteration or ensemble complexity, making it more suitable for real world deployment.

Overall, these results demonstrate that feature-level hybrid fusion provides a more effective and efficient alternative to both traditional models and complex ensemble strategies.

4.2.7. Class-Level Trade-offs

Our models exhibited clear precision recall trade-offs. The proposed hybrid achieved strong Hate Recall (0.7503), enabling effective detection of offensive content, while the MuRIL + XLM-R stacking model attained the peak Hate Precision (0.9296), making it suitable for moderation scenarios requiring minimal false positives.

4.2.8. Why TF-IDF Fusion Underperformed

The MuRIL + XLM-R + TF-IDF approach performed poorly (Macro-F1: 0.7004) due to the incompatibility between sparse TF-IDF features and dense transformer embeddings, which introduces lexical noise in morphologically rich Tamil text.

4.2.9. Summary

Overall, the results confirm that combining multi-layer transformer representations, dual pooling, and FastText embeddings outperforms both single-model baselines and traditional decision-level ensembles for Tamil hate speech detection.

5. Conclusion

This work evaluates transformer-based, hybrid, and ensemble architectures for Tamil hate speech detection, using Macro-F1 as the primary metric. The proposed L3Cube-TamilBERT + FastText hybrid framework, which combines deep contextual embeddings with FastText representations achieves F1 score of 0.7883 While stacking ensembles and multilingual hybrids in Table 2 achieve better hate precision and hate recall, they compromise overall stability showing low F1

scores. These results demonstrate that the proposed feature-fusion ensemble approach which integrate multi-layer representations with dual pooling offer a more reliable and scalable approach for low-resource languages like Tamil, particularly when benchmark datasets are limited or not publicly available.

6. Bibliographical References

References

Junita Amalia, Sarah Rosdiana Tambunan, Susi Eva Maria Purba, and Walker Valentinus Simanjuntak. 2025. Enhancing hate speech detection: Leveraging emoji preprocessing with bi-lstm model. *Journal of Information Systems and Informatics*, 7(2). Published June 2025.

Fazlourrahman Balouchzahi, Sepideh Bashang, Grigori Sidorov, and H. Shashirekha. 2022. Comata oli-code-mixed malayalam and tamil offensive language identification.

Sean Benhur and Kanchana Sivanraju. 2021. Psg@hasoc-dravidian codemixfire2021: Pretrained transformers for offensive language identification in tanglish. *CoRR*, abs/2110.02852.

Paula Fortuna and Sergio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys*.

A. Iftikhar et al. 2025. Beyond words: A hybrid transformer-ensemble approach for detecting hate speech and offensive language on social media. *PeerJ Computer Science*.

S. F. Karim et al. 2025. Cuet_blitz_aces@lt-edi-2025: Leveraging transformer ensembles and majority voting for hate speech detection. In *Proceedings of the 5th Workshop on Speech, Vision, and Language Technologies for Dravidian Languages (ACL Anthology)*.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2021. Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '20, page 29–32, New York, NY, USA. Association for Computing Machinery.

C Maria Nancy, N Radha, and R Swathika. 2025. SSN_IT_HATE@LT-EDI-2025: Caste and migration hate speech detection. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 84–89, Naples, Italy. Unior Press.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

B. Roy et al. 2025. Lexilogic@dravidianlangtech 2025: Multimodal hate speech detection in dravidian languages. In *Proceedings of the DravidianLangTech@NAACL 2025 (ACL Anthology)*.

Ganesh Sundhar S, Durai Singh K, Gnanasabesan G, Hari Krishnan N, and Mc Dhanush. 2025. Wise@LT-EDI-2025: Combining classical and neural representations with multi-scale ensemble learning for code-mixed hate speech detection. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 54–62, Naples, Italy. Unior Press.

Debjoy Saha, Naman Paharia, Debajit Chakraborty, Punyajoy Saha, and Animesh Mukherjee. 2021. Hate-alert@DravidianLangTech-EACL2021: Ensembling strategies for transformer-based offensive language detection. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 270–276, Kyiv. Association for Computational Linguistics.

Siva Sai and Yashvardhan Sharma. 2020. Siva@ hasoc-dravidian-codemix-fire-2020: Multilingual offensive speech detection in code-mixed and romanized text. In *FIRE (Working Notes)*, pages 336–343.

S. Sangeetham et al. 2024. Enhanced detection of hate speech in dravidian languages in social media using ensemble transformers. *Interdisciplinary Journal of Information, Knowledge, and Management*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. *CoRR*, abs/1905.05950.

Ning Xu. 2026. Tokenization and morphological fidelity in Uralic NLP. *arXiv preprint*. ArXiv:2602.04241.

(Mandl et al., 2021)

7. Language Resource References

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2021. Overview of the track on sentiment analysis for dravidian languages in code-mixed text. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '20, page 21–24, New York, NY, USA. Association for Computing Machinery.

Hata, Eshika Nahata. 2020. *Tamil Offensive Speech Detection*. Kaggle. Kaggle dataset; accessed 2026-02-27.

Asrita Venkata Mandalam, Yashvardhan Sharma, Bharathi Raja Chakravarthi, et al. 2020. Overview of the hasoc-dravidiancodemix shared task on offensive language identification in code-mixed text. In *Proceedings of the 12th Forum for Information Retrieval Evaluation (FIRE 2020)*, pages 55–59.

Comparative Analysis of Tokenizers in Tamil Text Classification in Low Resource Settings

Gokulan Sivakumaran[1], **Randil Pushpananda**[2], **ERAD Bandara**[1]
[1]Department of Statistics, University of Colombo, Sri Lanka
[2]University of Colombo School of Computing, University of Colombo, Sri Lanka
sivakumarangokulan@gmail.com, rpn@ucsc.cmb.ac.lk, anjana@stat.cmb.ac.lk

Abstract

Tokenization is crucial in NLP, influencing performance for morphologically rich, low resource languages like Tamil. This study comprehensively analyzes WordPiece, SentencePiece, and Byte-Level Byte Pair Encoding (BBPE) for Tamil text classification. We assess tokenization efficiency using metrics including token count, fragmentation, OOV rate, and compression ratio. Additionally, we analyze downstream impact through Tamil news title classification using a custom lightweight BERT based Transformer architecture. Tokenizers were pretrained on a 5.45 GB Tamil Corpus and evaluated on a Kaggle Tamil News Dataset. Results indicate WordPiece and SentencePiece outperform BBPE in efficiency and accuracy. While BBPE eliminates OOV words, excessive fragmentation hinders model learning. Increasing vocabulary size improves WordPiece and SentencePiece but not BBPE. Misclassification analysis highlights overfragmentation challenges. This study contributes to Tamil NLP by comparing tokenizers, aiding researchers in selecting appropriate strategies for agglutinative languages.

Keywords: Tokenization, Tamil, subword methods, low resource NLP, text classification

1. Introduction

Tokenization is a critical component of modern Natural Language Processing (NLP) systems, directly affecting vocabulary efficiency, semantic representation, and downstream task performance in transformer based models (Devlin et al., 2019; Wolf et al., 2020). Subword tokenization methods such as WordPiece, Byte Pair Encoding (BPE), and SentencePiece have become standard in high resource language settings (Wu et al., 2016; Sennrich et al., 2016; Kudo and Richardson, 2018a). These approaches effectively mitigate large vocabulary sizes and Out Of Vocabulary (OOV) issues while preserving compositional semantics.

However, their effectiveness in morphologically rich and low resource languages remains insufficiently understood. Tamil, an agglutinative Dravidian language, presents substantial challenges due to its extensive inflectional morphology, compound formation, and divergence between written and spoken forms (Annamalai and Asher, 2002; Thangarasu and Manavalan, 2013; Sarveswaran et al., 2021). When subword tokenizers designed primarily for English centric corpora are applied directly to Tamil, they often produce excessive fragmentation or linguistically inconsistent segmentations, potentially degrading semantic coherence and model performance.

Transformer architectures increasingly rely on subword tokenization to control vocabulary growth and improve generalization (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019). Prior research demonstrates that tokenization design can significantly impact representation quality and downstream performance (Bostrom and Durrett, 2020; Rust et al., 2020). Nevertheless, systematic evaluations across morphologically complex low resource languages remain limited. Existing Tamil NLP efforts largely emphasize preprocessing tools and rule based segmentation pipelines rather than empirical comparisons of modern subword tokenization strategies (Singh and Shah, 2022). Consequently, the relationship between tokenization granularity and downstream task performance in Tamil has not been rigorously examined.

To address this gap, we performed a comparative evaluation of three widely used subword tokenization methods: WordPiece, SentencePiece, and BBPE. We assess both intrinsic tokenization characteristics and extrinsic downstream performance using Tamil news title classification with custom lightweight BERT based Transformer architecture (Devlin et al., 2019). News headlines constitute a suitable benchmark because they are short, semantically dense, and require precise lexical segmentation for accurate categorization (Veeramanju, 2023).

This paper makes the following contributions:

- Assess and compare the tokenization efficiency of BBPE, WordPiece, and SentencePiece using various Tokenization Quality metrics.
- Investigate how effectively each tokenization approach preserves semantic meaning in Tamil text classification by analyzing the performance of the model on a downstream task, specifically news title classification.

Our findings highlight the importance of language

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 198–208
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

aware tokenization design and contribute to improving multilingual language modeling for underrepresented languages.

2. Related Work

2.1. Tokenization Strategies and Efficiency

The choice of tokenization algorithm critically impacts model performance, particularly in balancing semantic fidelity with computational efficiency. Qarah and Alsanoosay (2024) conducted a comparative analysis of WordPiece (WP), SentencePiece (SP), and Byte-Level BPE (BBPE) for Arabic NLP tasks. Their findings indicate that SP achieves stronger task performance by better modeling morphological variation, albeit at increased computational cost. In contrast, BBPE offers faster training and inference but suffers from semantic degradation due to excessive token fragmentation.

These trade-offs are further influenced by vocabulary size. Hiraoka et al. (2020) show that larger vocabularies improve representational capacity but increase memory and computation overhead, while Dagan et al. (2024) demonstrate that aggressive compression can limit semantic expressiveness. In low resource settings, where memory footprint and inference latency are constrained, selecting an appropriate tokenization strategy is therefore as critical as model architecture design.

2.2. Tokenization in Morphologically Rich Languages

For agglutinative and morphologically complex languages, subword tokenization has proven essential (Kudo and Richardson, 2018b; Rust et al., 2021). In Korean NLP, Park et al. (2020) demonstrate that a hybrid approach combining morphological analysis with subword segmentation (e.g., morpheme+BPE) often outperforms pure subword or pure morpheme tokenization across tasks including sentiment analysis.

Similarly, extensive research in Arabic NLP (Antoun et al., 2020; Abdul-Mageed et al., 2020; Abdelali et al., 2021) confirms that subword tokenization substantially improves performance across classification and sequence labeling tasks. However, these studies focus primarily on Semitic or Altaic language families.

Byte-level approaches, while language agnostic, introduce challenges in morphologically dense languages. Wei et al. (2021) show that byte-level subword modeling can generalize across scripts, but excessive fragmentation may obscure affixal and morphemic information, which is crucial for agglutinative languages such as those in the Dravidian family.

2.3. Research Gap

Despite extensive research on Arabic and Korean, there is a clear lack of systematic evaluations for Tamil, a low-resource and agglutinative Dravidian language. While foundational linguistic resources and NLP tools exist, few studies explicitly isolate the impact of tokenization strategies on both computational efficiency and downstream task performance.

Existing work often evaluates tokenization either through intrinsic efficiency metrics or downstream accuracy in isolation. This study addresses this gap by providing a holistic evaluation of WP, SP, and BBPE for Tamil, assessing intrinsic tokenization quality (e.g., fragmentation, compression, and vocabulary efficiency) alongside extrinsic performance on a news classification task. By establishing a unified evaluation framework, this work contributes toward more efficient and linguistically informed language models for underrepresented languages.

3. Background

This section introduces the core tokenization strategies and model architectures that form the foundation of this research. We briefly describe WordPiece, SentencePiece, and BBPE, followed by an overview of BERT based transformer models used for downstream evaluation.

3.1. WordPiece Tokenization

WordPiece is a subword tokenization algorithm originally proposed for neural machine translation and later widely adopted in transformer-based language models such as BERT (Wu et al., 2016; Devlin et al., 2019). The algorithm constructs a fixed size vocabulary by iteratively merging character sequences that maximize the likelihood of the training corpus. During tokenization, words are segmented into the longest matching subword units available in the learned vocabulary.

For example, the English sentence *"I was uncomfortable"* may be tokenized as: `[I]`, `[was]`, `[un]`, `[##comfortable]`, where the prefix `##` indicates that the token is a continuation of a preceding word. This strategy allows WordPiece to balance vocabulary coverage and semantic consistency while efficiently representing rare or unseen words through subword decomposition. The wordpiece tokenizer has been used in many prominent language models such as BERT, DistillBERT,and IndicBERT.

3.2. SentencePiece Tokenization

SentencePiece is an unsupervised and language independent subword tokenization framework that learns segmentation directly from raw text without requiring prior word level tokenization (Kudo and Richardson, 2018a; Kudo, 2018). Unlike traditional tokenizers that rely on whitespace splitting, SentencePiece treats input as a continuous stream of Unicode characters and learns subword units based on statistical properties of the training corpus.

For example, the English phrase *"natural language processing"* may be segmented into: `[_natural]`, `[_language]`, `[_process]`, `[ing]`, where the underscore symbol represents a whitespace boundary. SentencePiece supports both unigram language model tokenization and BPE based segmentation, allowing flexible handling of morphologically complex or multilingual text.The SentencePiece tokenizer has been used in many prominent language models such as ALBERT, mT5, XLNet , and T5.

3.3. Byte-Level Byte Pair Encoding (BBPE)

Byte-Level Byte Pair Encoding is a variant of the traditional BPE algorithm that operates directly on byte sequences rather than characters (Sennrich et al., 2016; Wei et al., 2021). By constructing a base vocabulary consisting of all possible byte values, BBPE guarantees complete text coverage and eliminates out-of-vocabulary tokens. We acknowledge that for languages like Tamil requiring multiple bytes per Unicode character in UTF-8 encoding, BBPE may introduce additional fragmentation compared to character-level BPE. However, BBPE was selected for this study for the following reasons:

1. **Standardization in Modern Architectures**: BBPE has become the default tokenization strategy in prominent transformer models such as GPT-2, RoBERTa, and GPT-3, making it a relevant baseline for contemporary NLP research (Radford et al., 2019; Liu et al., 2019).
2. **Cross-Lingual Comparability**: BBPE provides a language-agnostic approach that enables fair comparison across diverse scripts without requiring language-specific preprocessing (Wei et al., 2021).
3. **OOV Elimination**: The byte-level approach guarantees 0% OOV rate, which is valuable for low-resource settings where vocabulary coverage is critical.

3.4. BERT-based Transformer Models

Transformer-based language models rely on tokenized input sequences to learn contextual representations through self-attention mechanisms (Vaswani et al., 2017; Devlin et al., 2019). BERT and its variants employ bidirectional attention to capture contextual dependencies between tokens, making tokenization a critical component influencing model efficiency, representation quality, and downstream task performance.

Variants such as RoBERTa and ALBERT extend the original BERT architecture through improved training strategies and parameter efficiency (Liu et al., 2019; Lan et al., 2020). Since these models operate directly on token sequences, the choice of tokenization strategy significantly affects sequence length, semantic representation, and classification performance, particularly in morphologically rich languages.

4. Methodology

4.1. Software and Other Specifications

The experimental implementation was developed in Python, selected for its extensive ecosystem of machine learning libraries and strong community support. All experiments were executed on the High-Performance Computing (HPC) cluster to accommodate the substantial computational demands of model pretraining. The software environment was configured on Ubuntu 22.04 LTS. PyTorch as the primary deep learning frameworks for model development across pretraining and fine tuning stages. The Hugging Face ecosystem served as the cornerstone for NLP operations: the Transformers library provided model architectures (BertModel, BertForSequenceClassification), training utilities (Trainer, TrainingArguments), and tokenization (AutoTokenizer), while the Datasets library facilitated efficient data loading and management. It is important to note that

4.2. Model Pretraining

The objective of this study is to conduct a comparative analysis of three widely used subword tokenization strategies WordPiece, SentencePiece, and BBPE as base tokenizers for Tamil language models. The tokenizer pretraining was carried out using the Tamil Corpus Praveen Govi (2020) obtained from Kaggle.

The raw corpus underwent a preprocessing phase in which metadata, emojis, and non-Tamil scripts such as Hindi and Arabic were removed. Punctuation marks, numerals, symbols, and embedded English words were intentionally preserved to retain realistic linguistic structure. After preprocessing, the cleaned corpus size was approximately 5.45 GB.

The processed text corpus was loaded using the Hugging Face `datasets` library into a dataset

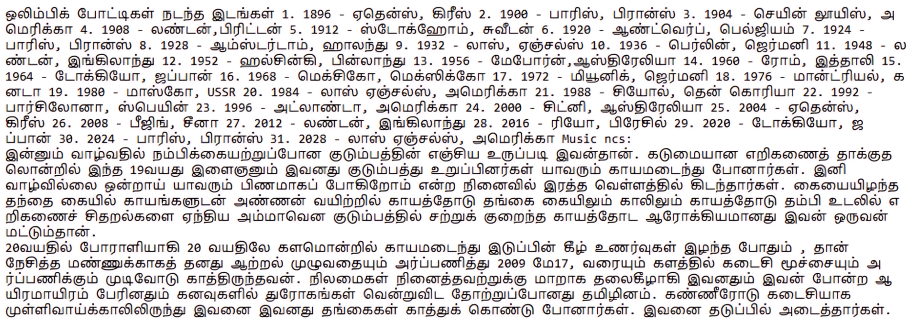

ஒலிம்பிக் போட்டிகள் நடந்த இடங்கள் 1. 1896 - ஏதென்ஸ், கிரீஸ் 2. 1900 - பாரிஸ், பிரான்ஸ் 3. 1904 - செயின் லூயிஸ், அ
மெரிக்கா 4. 1908 - லண்டன்,பிரிட்டன் 5. 1912 - ஸ்டோக்ஹோம், சுவீடன் 6. 1920 - ஆண்ட்வெர்ப், பெல்ஜியம் 7. 1924 -
பாரிஸ், பிரான்ஸ் 8. 1928 - ஆம்ஸ்டர்டாம், ஹாலந்து 9. 1932 - லாஸ், ஏஞ்சல்ஸ் 10. 1936 - பெர்லின், ஜெர்மனி 11. 1948 - ல
ண்டன், இங்கிலாந்து 12. 1952 - ஹல்சின்கி, பின்லாந்து 13. 1956 - மேபோர்ன்,ஆஸ்திரேலியா 14. 1960 - ரோம், இத்தாலி 15.
1964 - டோக்கியோ, ஜப்பான் 16. 1968 - மெக்சிகோ, மெக்ஸிக்கோ 17. 1972 - மியூனிக், ஜெர்மனி 18. 1976 - மான்ட்ரியல், க
னடா 19. 1980 - மாஸ்கோ, USSR 20. 1984 - லாஸ் ஏஞ்சல்ஸ், அமெரிக்கா 21. 1988 - சியோல், தென் கொரியா 22. 1992 -
பார்சிலோனா, ஸ்பெயின் 23. 1996 - அட்லாண்டா, அமெரிக்கா 24. 2000 - சிட்னி, ஆஸ்திரேலியா 25. 2004 - ஏதென்ஸ்,
கிரீஸ் 26. 2008 - பீஜிங், சீனா 27. 2012 - லண்டன், இங்கிலாந்து 28. 2016 - ரியோ, பிரேசில் 29. 2020 - டோக்கியோ, ஜ
ப்பான் 30. 2024 - பாரிஸ், பிரான்ஸ் 31. 2028 - லாஸ் ஏஞ்சல்ஸ், அமெரிக்கா Music ncs:
இன்னும் வாழ்வதில் நம்பிக்கையற்றுப்போன குடும்பத்தின் எஞ்சிய உருப்படி இவன்தான். கடுமையான எறிகணைத் தாக்குத
லொன்றில் இந்த 19வயது இளைஞனும் இவனது குடும்பத்து உறுப்பினர்கள் யாவரும் காயமடைந்து போனார்கள். இனி
வாழ்வில்லை ஒன்றாய் யாவரும் பிணமாகப் போகிறோம் என்ற நினைவில் இரத்த வெள்ளத்தில் கிடந்தார்கள். கையையிழந்த
தந்தை கையில் காயங்களுடன் அண்ணன் வயிற்றில் காயத்தோடு தங்கை கையிலும் காலிலும் காயத்தோடு தம்பி உடலில் எ
றிகணைச் சிதறல்களை ஏந்திய அம்மாவென குடும்பத்தில் சற்றுக் குறைந்த காயத்தோட ஆரோக்கியமானது இவன் ஒருவன்
மட்டும்தான்.
20வயதில் போராளியாகி 20 வயதிலே களமொன்றில் காயமடைந்து இடுப்பின் கீழ் உணர்வுகள் இழந்த போதும் , தான்
நேசித்த மண்ணுக்காகத் தனது ஆற்றல் முழுவதையும் அர்ப்பணித்து 2009 மே17, வரையும் களத்தில் கடைசி மூச்சையும் அ
ர்ப்பணிக்கும் முடிவோடு காத்திருந்தவன். நிலமைகள் நினைத்தவற்றுக்கு மாறாக தலைகீழாகி இவனதும் இவன் போன்ற ஆ
யிரமாயிரம் பேரினதும் கனவுகளில் துரோகங்கள் வென்றுவிட தோற்றுப்போனது தமிழினம். கண்ணீரோடு கடைசியாக
முள்ளிவாய்க்காலிலிருந்து இவனை இவனது தங்கைகள் காத்துக் கொண்டு போனார்கள். இவனை தடுப்பில் அடைத்தார்கள்.

Figure 1: Snapshot of the Tamil Corpus

containing a single column named `text`. To ensure a fair comparison across tokenizers, a controlled experimental setup was adopted in which the model architecture remained constant while only the tokenization strategy differed. We trained three subword tokenizers from scratch, initializing their algorithm configurations based on standard pretrained models:

WordPiece (WP): Configuration initialized from `bert-base-uncased`.

Byte-Level BPE (BBPE): Configuration initialized from `roberta-base`.

SentencePiece (SP): Configuration initialized from `albert-base-v1`.

Note: These references indicate the tokenization algorithm type only; no pretrained vocabulary or weights were transferred.

For the language model, we employed a custom lightweight Transformer architecture for all three settings with 3 encoder layers, 3 attention heads per layer, and a hidden layer size of 192 units,and the vocabulary size was set to 20,000. Furthermore, we used the following configurations for pretraining the models: masking 15% of the input tokens, a maximum sequence length of 64, a batch size of 64, and the 'AdamW' optimizer, with a learning rate of 0.0001. In addition, a Word Level tokenizer is pretrained and employed as a control baseline to facilitate comparison with subword based tokenization methods.

4.3. Finetuning for Downstream Task

For the training of Tamil news classification models, a structured procedure was followed to ensure reproducibility. The process began by setting a fixed seed value for consistency across runs. The tokenizer and model were then loaded using the pretrained weights. The tokenizer was utilized to preprocess the dataset by truncating and padding input sequences to 64 tokens.

For the downstream classification task, we employed a BERT-based transformer architecture. This choice aligns with recent comparative tokenization studies in morphologically rich languages, such as (Qarah and Alsanoosay, 2024), which standardized on `BertForSequenceClassification` to isolate tokenization variables across Arabic NLP tasks. While we acknowledge potential architecture-tokenizer affinity (e.g., WordPiece being native to BERT), we conducted preliminary validation using `AlbertForSequenceClassification` and `RobertaForSequenceClassification` models. These experiments confirmed that the relative performance ranking of tokenizers (WordPiece > SentencePiece > BBPE) remains consistent across architectures, indicating that the observed performance gaps are driven by tokenization quality rather than model compatibility alone. Consequently, we report results using the BERT architecture, which yielded the highest overall accuracy among the three classifiers tested.

For evaluation, accuracy and F1-score were computed. In addition, misclassified samples were identified and logged for qualitative analysis. A set of training arguments was configured, including 5 epochs, batch size 64, and learning rate 0.0001. Mixed precision training (fp16) was enabled. The training process was executed using the Hugging Face Trainer API. The procedure was carried out 5 times setting different seeds to obtain an average performance.

4.4. Dataset

The dataset used for the downstream task is obtained from Kaggle Vijaya Bhaskar (2020) which is a collection of Tamil news data collected from the Tamilmurasu website. When observing the count of news categories, which is in the news category column in the dataset, a high class imbalance was observed as shown in Figure 2. Since the prediction of BERT models is affected by class imbalance (Madabushi et al., 2019), the dataset needed to be balanced. To achieve this, categories whose counts were greater than 7000 were selected for

	news_category	Count
0	ஆன்மீகம்	406
1	இந்தியா	16935
2	உலகம்	7477
3	கல்வி	240
4	குற்றம்	16290
5	சினிமா(ரீல்மா)	9248
6	தமிழகம்	53333
7	தலையங்கம்	1535
8	தொழில்	68
9	மருத்துவம்	544
10	மர்மம்	66
11	மாவட்ட மசாலா	9079
12	விளையாட்டு	8230
13	வேலைவாய்ப்பு	1042
14	ஸ்டேட் எக்ஸ்பிரஸ்	2253

Figure 2: Snapshot of the count of News Categories

the final dataset. Figure 3 shows the news categories and their distribution in the balanced dataset. The dataset was split into 80% for training and 20% for testing. To maintain the same balanced distribution in the train set, the stratify option of `train_test_split` was used.

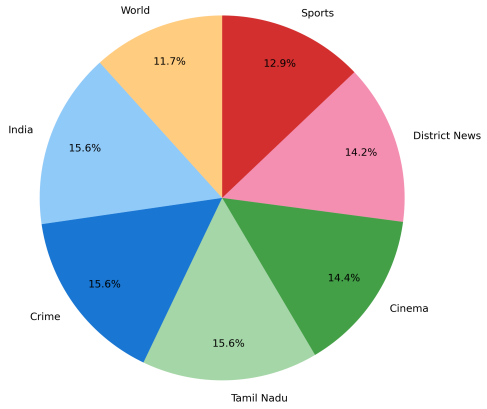


Figure 3: Piechart for distribution of News Categories after balancing

The news titles column served as the input source for both testing tokenization quality metrics and downstream classification tasks, where instances are categorized into one of several predefined categories. An example of the tokenization of a text from the news titles using the pretrained models is shown in figure 4.

4.5. Evaluation Metrics

To evaluate the performance of the tokenizers and the downstream model, we employ the following metrics.

4.5.1. Tokenization Metrics

Token Count per Sentence (TCS) This metric computes the average number of tokens produced per sentence after tokenization:

$$\text{Avg. Token Count} = \frac{1}{N}\sum_{i=1}^{N} T_i \tag{1}$$

where N is the total number of sentences and T_i denotes the number of tokens in the i-th sentence. A higher token count indicates frequent subword splitting.

Subword Fragmentation Score (SFS) This measures how often words are split into multiple subwords:

$$\text{Frag. Score} = \frac{\text{No. of Subword Tokens}}{\text{No. of Original Words}} \tag{2}$$

A higher score reflects excessive word breaking, which may negatively affect language understanding.

Out-of-Vocabulary Rate (OOVR) The OOV rate measures how frequently tokens are mapped to the unknown token (`<unk>`):

$$\text{OOV Rate} = \frac{\text{No. of OOV Tokens}}{\text{Total No. of Tokens}} \tag{3}$$

A lower OOV rate indicates better vocabulary coverage.

Compression Ratio (CR) This evaluates how efficiently the tokenizer represents the input text:

$$\text{Comp. Ratio} = \frac{\text{Original Seq. Length}}{\text{Tokenized Seq. Length}} \tag{4}$$

Here, lengths refer to word count and token count, respectively. A higher ratio implies more efficient tokenization.

4.5.2. Classification Metrics

Accuracy. Accuracy measures the proportion of correctly classified samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives.

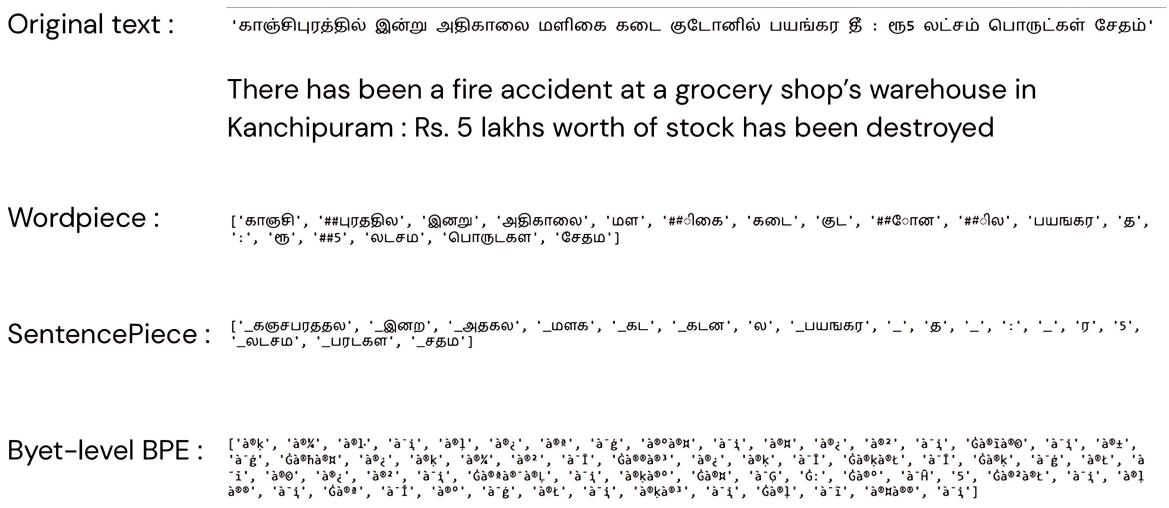


Figure 4: Snapshot of the example of the tokenization of a text

F1 Score. The F1 score is the harmonic mean of precision and recall. To save space, we define Precision (P) and Recall (R) inline: $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$. The F1 score is then:

$$\text{F1} = \frac{2 \times P \times R}{P + R} \tag{6}$$

This metric is particularly useful for evaluating performance on imbalanced datasets.

5. Results and Discussion

Metric	WP	SP	BBPE
TCS	11.332	**10.353**	45.503
SFS	1.529	**1.381**	6.115
OOVR	0.060	0.059	**0.000**
CR	5.330	**5.835**	1.328

Table 1: Tokenization Quality Metrics Results

From Table 1, Our comparative analysis demonstrates clear distinctions among Byte-Level BPE (BBPE), WordPiece (WP), and SentencePiece (SP) across multiple evaluation metrics. BBPE produces substantially more tokens per sentence (45.503) relative to WordPiece (11.332) and SentencePiece (10.353), reflecting its tendency to decompose words into fine grained subword units. This behavior is corroborated by the subword fragmentation score, where BBPE registers 6.115 fragments per word, compared to 1.529 for WordPiece and 1.381 for SentencePiece. The higher fragmentation in BBPE yields longer input sequences, which can increase memory consumption and slow both training and inference.

Compression ratio and token overlap metrics reinforce these observations. BBPE attains the lowest compression ratio (1.328), confirming that its tokenized output expands significantly relative to the original text. In contrast, WordPiece (5.330) and SentencePiece (5.835) achieve markedly higher compression, indicating greater information density per token.

Taken together, these results indicate that BBPE prioritizes universal coverage at the expense of sequence length, vocabulary diversity, and computational efficiency. WordPiece and SentencePiece strike a more favorable balance for downstream classification tasks, offering compact representations, manageable vocabulary sizes, and reasonable out of vocabulary handling. For resource constrained settings or architectures sensitive to input length, the latter two methods present a more practical choice without substantially sacrificing lexical coverage.

Metrics	Word level	WP	SP	BBPE
Accuracy	72.86%	**75.55%**	74.32%	56.88%
F1 Score	72.93%	**75.58%**	74.36%	54.64%

Table 2: Train set results classification with 20000 vocabulary

Metrics	Word level	WP	SP	BBPE
Accuracy	67.76%	**72.42%**	70.73%	50.66%
F1 Score	67.51%	**72.34%**	70.55%	48.97%

Table 3: Test set results classification with 20000 vocabulary

From the Table 2 and 3 results, it can be observed that WordPiece tokenization consistently yields the highest performance across both training and test sets. It achieves 72.42% accuracy and 72.34% F1 score on the test set. SentencePiece closely follows. The Word-Level baseline shows modest performance, while Byte-Level BPE performs significantly worse, especially on the test set where it achieves only 50.66% accuracy, indicating poor generalization.

The confusion matrix shown in Figures 5, 6, and 7 illustrates how each category is predicted by the tokenizers in the test set. The description of the numbers that represent each category is Cinema: 0, Crime: 1, District News: 2, India: 3, Sports: 4, Tamil Nadu: 5, World: 6

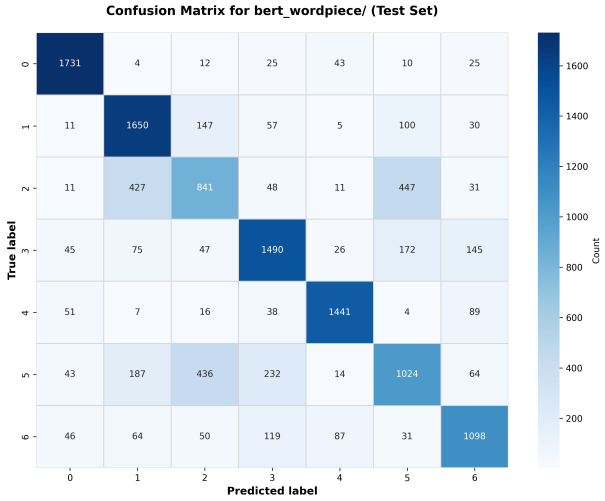


Figure 5: Confusion matrix for WordPiece

A qualitative error analysis was conducted on both misclassified and correctly classified samples to comprehensively understand the relationship between tokenization and classification performance. For misclassified samples, errors primarily stemmed from over-fragmentation and under-fragmentation in WordPiece and SentencePiece, particularly in the case of compound or morphologically rich words. Additionally, both subword methods occasionally segmented named entities (such as city or person names), which, while not inherently problematic for classification, can degrade model performance when such segments lose semantic meaning. However, to ensure these patterns are indeed correlated with classification errors rather than occurring universally, we also examined correctly classified samples. Our analysis revealed that properly classified instances generally exhibited appropriate tokenization boundaries without excessive over-fragmentation or under-fragmentation. This comparative analysis confirms that tokenization quality directly influences downstream classification accuracy. Figures 8 and 9 illustrate the over-fragmentation and under-fragmentation issues observed in misclassified samples for WordPiece and SentencePiece tokenization, respectively. In contrast, Figures 10 and 11 demonstrate proper tokenization with correct classification for WordPiece and SentencePiece, showing well-formed token boundaries that preserve semantic meaning. These patterns suggest tokenization errors can propagate downstream and affect classification accuracy, while appropriate tokenization supports reliable model performance.

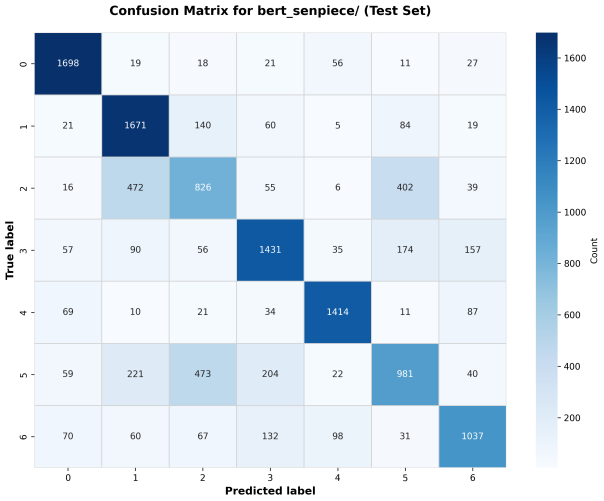


Figure 6: Confusion matrix for Sentencepiece

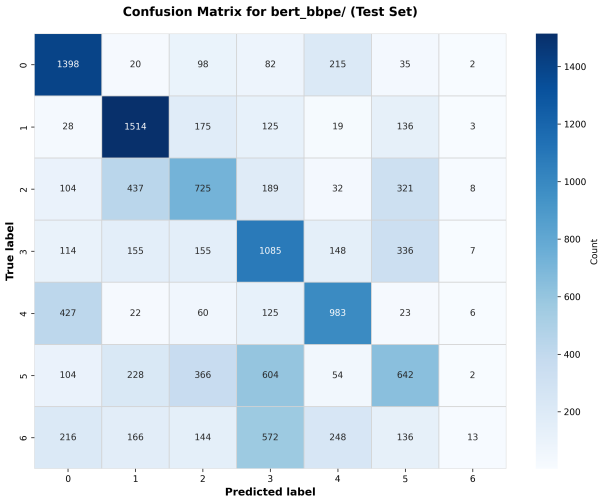


Figure 7: Confusion matrix for Byte-Level BPE

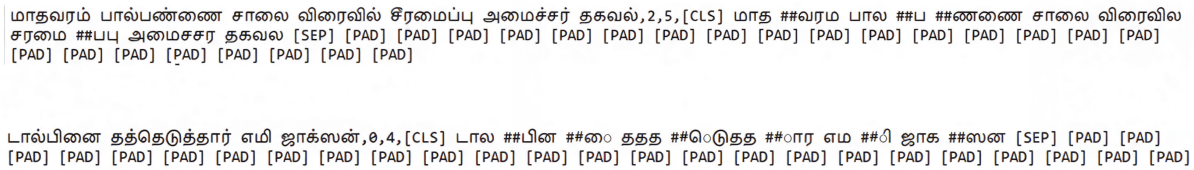


Figure 8: WordPiece Segmentation in misclassification

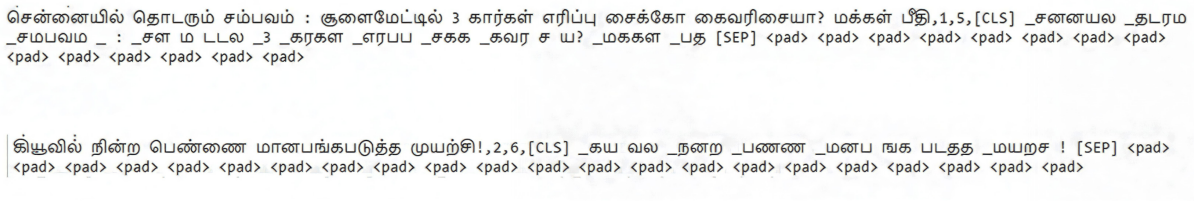


Figure 9: SentencePiece Segmentation in misclassification

In contrast, analyzing errors for BBPE proved more complex. Unlike WordPiece and SentencePiece, BBPE outputs encoded byte sequences, which are not readily interpretable in human-readable subwords. Decoding was necessary to reconstruct the original input for meaningful analysis.

To improve the accuracy of the tokenizers, particularly WordPiece and SentencePiece, which exhibited tokenization issues, they were pretrained with a larger vocabulary of 50,000. This adjustment is focused to address fragmentation problems and enhance their tokenization efficiency. Additionally, BBPE was also pretrained with a 50,000 vocabulary size to evaluate whether increasing the vocabulary would lead to any significant changes in classification performance. The results of the classification task after pretraining each tokenizer with a 50,000 vocabulary size are presented in Table 4 and 5.

Metrics	WP	SP	BBPE
Accuracy	**76.19%**	75.07%	56.30%
F1 Score	**76.23%**	75.10%	54.35%

Table 4: Train set results classification with 50000 vocabulary

Metrics	WP	SP	BBPE
Accuracy	**73.05%**	71.69%	50.24%
F1 Score	**72.96%**	71.57%	47.35%

Table 5: Test set results classification with 50000 vocabulary

The results show a modest improvement for both WordPiece and SentencePiece tokenizers when the vocabulary size is increased. WordPiece reached a peak training accuracy of 76.19% and test accuracy of 73.05%, suggesting enhanced learning and better generalization. SentencePiece also showed slight improvements in both training and test metrics. However, BBPE did not benefit from the larger vocabulary. In fact, its performance slightly declined in the training set and remained almost unchanged in the test set, with only 50.24% accuracy and 47.35% F1 score. An additional attempt was made to improve BBPE's performance by applying a pre-tokenization strategy that first split the input based on whitespace before feeding it to the BBPE model. Despite this modification, there was no significant improvement in its classification metrics.

6. Conclusion

This study presents a systematic empirical evaluation comparing tokenization strategies for Tamil language modeling using both intrinsic (tokenization quality) and extrinsic (news classification) assessments. The findings reinforce that subword-based tokenization, especially WordPiece (72.42%) and SentencePiece (70.73%), can be effective for Tamil in the Text Classification task and also maintain efficiency in tokenization. While BBPE excels in eliminating OOV words, its over-fragmentation hampers semantic modeling and Text Classification downstream performance (50.66%). The practical takeaway is that tokenizer choice significantly impacts NLP outcomes and must balance efficiency, semantic preservation, and model compatibility. By understanding the strengths and weaknesses of each tokenizer, future Tamil NLP applications can be better equipped to handle the challenges of agglutinative morphology and data scarcity.

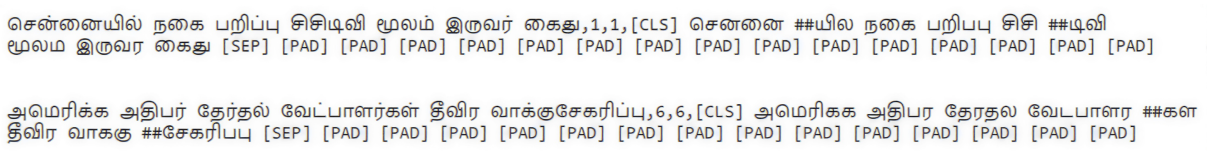


Figure 10: WordPiece Segmentation in correct classification

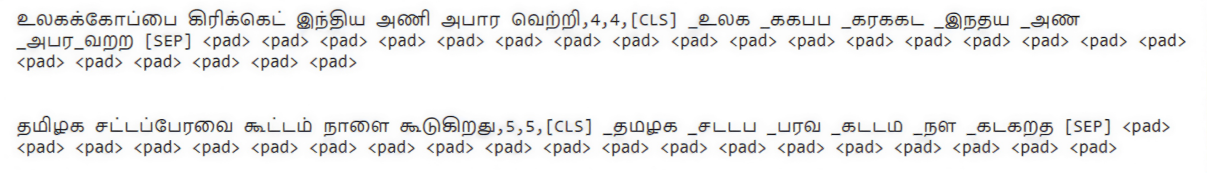


Figure 11: SentencePiece Segmentation in correct classification

7. Limitations

This study faced several limitations, primarily stemming from computational constraints, which concurrently outline the trajectory for future research. Specifically, a larger Tamil corpus for pretraining was not utilized, suggesting that future work should focus on leveraging extensive Tamil corpora to enhance representation learning. Furthermore, the maximum sequence length was restricted to 64 tokens; subsequent research could benefit from increasing this length to capture longer contextual dependencies. Due to limited GPU resources, simpler transformer architectures were employed, indicating that exploring deeper transformer models and hybrid tokenization approaches remains a key avenue for improvement. For Tamil characters requiring multiple bytes in UTF-8, byte-level approaches may introduce additional overhead. Character-level BPE should be evaluated to isolate encoding effects from tokenization algorithm effects. Finally, while the evaluation was confined to Text Classification due to the short sequence length's inability to capture longer sequences, applying tokenization analysis to diverse tasks such as sentiment analysis and Named Entity Recognition (NER) with higher sequence length like 128 would provide a more robust assessment of the model's generalizability.

8. Ethical Considerations

The Tamil Corpus and Tami News dataset is publicly available on Kaggle under standard licensing. All code and trained models is publicly available on https://github.com/Gokul200089/Tokenizers. Experiments were run on shared HPC resources; we minimized carbon footprint by using efficient lightweight architectures. Limitations regarding dialectal coverage are discussed in Section 7.

9. Bibliographical References

Ahmed Abdelali et al. 2021. Arabic pre-trained language models. In *Proceedings of the Sixth Arabic NLP Workshop*.

Muhammad Abdul-Mageed et al. 2020. Camelbert: Surprising cross-lingual transfer results. In *Proceedings of EMNLP 2020*.

E. Annamalai and R. E. Asher. 2002. *Colloquial Tamil: The Complete Course for Beginners*. Routledge.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of the Fourth Workshop on Open-Source Arabic Corpora and Processing Tools*.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of EMNLP 2020*, pages 4145–4155.

Guy Dagan et al. 2024. Tokenization from a compression and semantic trade-off perspective. *Transactions of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. 2020. Optimizing word segmentation for downstream task. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1341–1351, Online. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 66–75.

Taku Kudo and John Richardson. 2018a. Sentencepiece: A simple and language independent subword tokenizer. In *Proceedings of EMNLP 2018: System Demonstrations*, pages 66–71.

Taku Kudo and John Richardson. 2018b. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, et al. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, et al. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Harish Tayyar Madabushi, Elena Kochkina, and Mathieu Castelle. 2019. Cost-sensitive bert for generalisable sentence classification on imbalanced data. In *Proceedings of the Second Workshop on NLP for Internet Freedom*, pages 125–134.

Kyubyong Park, Joohong Lee, Seongbo Jang, and Dawoon Jung. 2020. An empirical study of tokenization strategies for various korean nlp tasks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 133–142, Suzhou, China. Association for Computational Linguistics.

Muhammad Qarah and Ibrahim Alsanoosay. 2024. A comparative study on tokenizers for arabic nlp tasks. In *Proceedings of the Conference on Arabic Natural Language Processing*.

Alec Radford, Jeffrey Wu, Rewon Child, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report*.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulic, et al. 2020. How good is your tokenizer? *arXiv preprint arXiv:2012.15613*.

K. Sarveswaran, G. Dias, and M. Butt. 2021. Thamizhi morph: A morphological parser for the tamil language. *Machine Translation*, 35(1):37–70.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725.

Harshit Singh and Rajiv Ratn Shah. 2022. Tamilnlp: Low resource language processing.

M. Thangarasu and R. Manavalan. 2013. Stemmers for tamil language: Performance analysis. *International Journal of Computer Trends and Technology*, 4(8):2582–2584.

Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

K. T. Veeramanju. 2023. A systematic review on news headlines categorization in malayalam. *Scope*, 13.

Jia Wei, Qiang Liu, Yiqun Guo, and Xia Jiang. 2021. Training multilingual pre-trained language models with byte-level subwords. *arXiv preprint arXiv:2101.09469*.

Thomas Wolf, Lysandre Debut, Victor Sanh, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45.

Yonghui Wu, Mike Schuster, Zhifeng Chen, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

10. Language Resource References

Praveen Govi. 2020. *Tamil Language Corpus for NLP*. Kaggle Datasets. Kaggle. A large collection of Tamil text corpora for natural language processing research.

Vijaya Bhaskar. 2020. *Tamil News Classification Dataset (Tamil Murasu)*. Kaggle Datasets. Kaggle. News articles collected from the Tamil Murasu newspaper and released for text classification research.

Improving Public Health Safety in Low-Resource Languages Using a Human-Verified Health Misinformation Corpus and Large Language Models

Sujal Maharjan[1,*], Astha Shrestha[1,*], Laxmi Thapa[2], Sweta Poudel[3], Shuvam Shiwakoti[4], Rabin Thapa[1], Kritesh Rauniyar[5], Surendrabikram Thapa[4]

[1]IIMS College, Kathmandu, Nepal, [2]O.P. Jindal Global University, India
[3]Kathmandu Engineering College, Tribhuvan University, Kathmandu, Nepal
[4]Virginia Tech, USA, [5]Macquarie University, Australia
{sujalmaharjan007, aasthashrestha688}@gmail.com

Abstract

The proliferation of health misinformation in Low-Resource Languages (LRLs) poses a severe threat to public health, yet automated detection remains critically under-studied due to the scarcity of high-quality benchmarks. We address this gap by introducing Nep-Health-Misinfo, a novel human-verified corpus for health misinformation identification in Nepali. The dataset was developed by adapting four foundational benchmarks (Monkeypox-V1, Monkeypox-V2, COVID-19, and CoAID) through a systematic Machine Translation Post-Editing (MTPE) protocol involving native experts. Our evaluation of Neural Machine Translation (NMT) systems reveals a significant translation asymmetry: while state-of-the-art (SOTA) systems achieve a BLEU score of 43.21 on factual health data, performance degrades sharply on deceptive narratives, with BLEU and TER scores dropping to 19.11 and 62.42, respectively. To establish robust baselines, we benchmark seven recent open-weight Large Language Models (LLMs), including Qwen2.5-7B-Instruct, Gemma-3-4B-IT, and Ministral-8B-Instruct, across zero-shot and few-shot settings. For the few-shot evaluation, we compare stochastic sampling against a K-means centroid-based approach for semantically representative exemplar selection. Experimental results indicate that Qwen2.5-7B-Instruct achieves a peak Macro F1-score of 0.8488, improving over its zero-shot performance (0.7188) on the same dataset. Our findings demonstrate that while few-shot prompting effectively mitigates distribution shifts in low-resource medical contexts, performance remains highly sensitive to the semantic density of exemplars. This work provides the first human-verified Nepali health misinformation corpus. All code and resources are available at https://github.com/SUJAL390/Nep-Health-Misinfo-CHIPSAL-LREC.

Keywords: Nepali NLP, health misinformation, large language models, machine translation post-editing

1. Introduction

The digital revolution has fundamentally restructured the ecosystem of health information dissemination, shifting the locus of authority from centralized medical specialists to decentralized, peer-to-peer online platforms (Swire-Thompson and Lazer, 2020; Eysenbach et al., 2008). While this democratization of data enables the rapid sharing of life-saving medical knowledge, it simultaneously provides a frictionless medium for the unchecked proliferation of health misinformation. The World Health Organization (WHO) has characterized this phenomenon as a global infodemic, a state where an overabundance of information, inclusive of both factual and deceptive narratives, complicates the public's ability to identify trustworthy guidance (Zarocostas, 2020; Cinelli et al., 2020). In high-stakes health domains, the cost of this informational decay is tangible: false claims regarding therapeutic efficacy, vaccine safety, and disease etiology have been dirdectly linked to treatment delays, increased vaccine hesitancy, and heightened mortality rates across diverse populations.

Despite the global nature of this crisis, a significant digital language divide persists in the development and deployment of automated misinformation detection systems (Joshi et al., 2020). Recent advances in Large Language Models (LLMs) have illustrated an exceptional proficiency in recognizing factual inconsistencies and adversarial narratives within high-resource languages such as English (Thapa et al., 2024; Zhou et al., 2023; Arora et al., 2025). However, these capabilities often fail to generalize to low-resource languages (LRLs), where a lack of high-quality, human-verified benchmarks hampers the fine-tuning and evaluation of safety protocols. Nepali, a language spoken by over 30 million people in the South Asian region, remains critically underserved in this regard. The scarcity of localized health corpora leaves Nepali speakers uniquely vulnerable to deceptive health narratives that exploit specific regional linguistic nuances and cultural contexts that global models, optimized primarily for Western medical corpora, frequently overlook (Alber et al., 2025; Acharya, 2023).

* The authors contributed equally to this work and are designated as joint first authors. The author order follows alphabetical order by last name.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 209–218
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

Table 1: Comparative Summary of Benchmarking and Misinformation Tasks. EN = English, NE = Nepali. vLMs = Vision-Language Models; QA = Question Answering.

Work	Task	Datasets	Model	Language
Zhang et al. (2025)	Health Misinfo	1	vLMs	EN
Maheshwari et al. (2025)	General QA	1	LLMs	11 Indic
Thapa et al. (2024)	Health Misinfo	4	LLMs	EN
Singh et al. (2024)	Multilingual Gen	1	LLMs	29 Indic
Ghimire and Shrestha (2025)	Fake News	1	XLM-R	NE/EN
Ours	**Health Misinfo**	**4**	**LLMs**	**NE**

A primary technical impediment to bridging this resource gap is the phenomenon of Translation Asymmetry. While modern Neural Machine Translation (NMT) systems like IndicTrans2 (Gala et al., 2023) have significantly improved general-purpose translation for Indic languages, their robustness degrades sharply when encountering the deceptive, grammatically non-standard, or adversarial linguistic patterns characteristic of misinformation. This decay in translation fidelity introduces semantic noise, where the underlying factual inconsistencies of a health claim are obscured by linguistic artifacts, ultimately degrading the downstream performance of detection frameworks. Consequently, simply translating English datasets into LRLs via automated means is insufficient for meticulous benchmarking; it necessitates a human-centric approach to ensure medical and cultural alignment. To address these challenges, this study introduces the Nep-Health-Misinfo corpus, a high-quality dataset for health misinformation identification in Nepali. We curated this corpus by adapting four foundational English benchmarks including Monkeypox-V1/V2 (Crone, 2022), COVID-19 (Patwa et al., 2021), and CoAID (Cui and Lee, 2020). These specific datasets were selected following the framework established by Thapa et al. (2024) and were processed utilizing a highly intensive Machine Translation Post-Editing (MTPE) protocol. By employing two native Nepali annotators with medical context awareness, we ensure that the dataset preserves the deceptive content of the original claims while reflecting natural Nepali syntax. This process allows us to quantify for the first time the translation gap between factual and deceptive health text in a low-resource environment.

Building upon the benchmarking framework established by Thapa et al. (2024), we evaluate seven recent open-weight LLMs: Qwen2.5-7B-Instruct (Qwen2.5-7B), Gemma-3-4B-IT (Gemma-3-4B), Ministral-8B-Instruct (Ministral-8B), Falcon3-7B-Instruct (Falcon3-7B), Phi-4-Mini-Instruct (Phi-4-Mini), Mistral-7B-Instruct-v0.3 (Mistral-7B), and Falcon-H1-3B-Instruct (Falcon-H1-3B). Following their experimental setup, our evaluation spans zero-shot and few-shot settings, comparing random sampling against a K-means centroid-based approach for selecting semantically representative exemplars. Our findings indicate that few-shot settings outperform zero-shot baselines. The contributions of this work are threefold: (i) we provide the Nep-Health-Misinfo corpus, the first human-verified health misinformation dataset for Nepali; (ii) we present a detailed technical analysis of how translation asymmetry affects the quality of low-resource health resources; and (iii) we establish a performance baseline for recent open-weight LLMs in a South Asian LRL context.

2. Related Work

The evolution of the digital landscape has facilitated an unprecedented shift in how medical knowledge is consumed and contested. Historically, research into health misinformation focused on the transition from centralized medical authority to decentralized, algorithmically driven information ecosystems (Swire-Thompson and Lazer, 2020). As Generative Artificial Intelligence (GenAI) matures, the challenge has shifted from identifying simple deceptive patterns to addressing sophisticated AI-driven narratives that mimic human-level credibility and expert reasoning (Zhou et al., 2023; Vraga and Bode, 2020).

2.1. Computational Paradigms in Detection

Automated misinformation detection has historically been grounded in heuristic-systematic psychological frameworks (Metzger, 2007). Early computational efforts relied on feature engineering and classical Machine Learning (ML) methods, such as Support Vector Machines (SVM) and Decision Trees, to classify textual patterns.

The introduction of the Transformer architecture marked a definitive paradigm shift from these manual feature-based approaches. Pre-trained Language Models (PLMs) like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) utilized

self-attention mechanisms to capture latent semantic relationships, significantly outperforming earlier methods (Devlin et al., 2019; Liu et al., 2019). Consequently, research during the COVID-19 pandemic largely transitioned toward these deep semantic models, though classical ML methods remained relevant for establishing baselines on static claims (Bojjireddy et al., 2021).

In the South Asian context, the research landscape remains largely anchored in this discriminative stage. Giri et al. (2024) utilized Convolutional Neural Networks (CNNs) to explore deceptive linguistic patterns in Nepali digital media, while Ghimire and Shrestha (2025) introduced bilingual classification frameworks utilizing XLM-RoBERTa for cross-lingual news classification. Despite their efficacy, these models are primarily classification-oriented and lack the zero-shot reasoning capabilities required to disentangle the nuanced factual inconsistencies found in modern medical misinformation.

2.2. LLM Safety and Hallucination Gap

The advent of LLMs has introduced Chain-of-Thought (CoT) reasoning, allowing models to perform multi-step fact-checking (Wei et al., 2022). Recent evaluations demonstrate that frontier models can surpass human benchmarks on medical exams; however, this performance is often deceptive. Huang et al. (2025) and Alber et al. (2025) highlight the hallucination gap, where models generate plausible but clinically dangerous advice due to statistical biases in their training data.

Global safety benchmarks have attempted to quantify these risks. Arora et al. (2025) introduced HealthBench, which employs physician-defined rubrics to evaluate clinical safety, and Thapa et al. (2024) established a critical framework for evaluating LLM susceptibility to medical myths. Furthermore, Zhang et al. (2025) developed MM-Health, a multimodal dataset illustrating how AI-generated health content can deceive existing state-of-the-art (SOTA) classifiers. A structured comparison of these benchmarking efforts is presented in Table 1. As evident from this comparison, existing benchmarks remain predominantly English-centric, leaving the safety profile of LLMs in LRLs largely unmapped.

2.3. Linguistic Resource Scarcity

A significant linguistic gap persists, as the majority of safety-critical research excludes languages like Nepali. Recent initiatives like IndicParam (Maheshwari et al., 2025) and IndicGenBench (Singh et al., 2024) have begun benchmarking South Asian languages for general tasks, but they lack domain-specific medical depth. A major bottleneck in LRLs research is the unreliability of NMT; as noted by Kamath et al. (2025), raw translation often distorts the medical intent and cultural nuance of health claims.

Our work addresses these challenges by introducing the Nep-Health-Misinfo corpus, a human-verified dataset developed via a highly intensive MTPE protocol. By extending the established experimental framework of Thapa et al. (2024) to this novel corpus, we provide the first comprehensive analysis of how recent open-weight models handle the semantic complexities of health misinformation in a South Asian LRLs environment. This study connects local discriminative methods with broader generative safety benchmarking, establishing a foundational baseline for public health safety in the region.

3. Dataset and Methodology

A critical contribution of this work is the creation of a high-quality Nepali dataset. To safeguard against the semantic drift common in low-resource translation, we employed a MTPE workflow to ensure semantic fidelity and cultural appropriateness.

3.1. Language Resource Construction

We introduce Nep-Health-Misinfo, a human-verified corpus designed to serve as a high-quality benchmark for health misinformation detection in the Devanagari script. The dataset comprises 4,924 unique instances aggregated from four established English datasets: CoAID, COVID-19, Monkeypox-V1, and Monkeypox-V2. Prior to translation, we applied a thorough data-cleaning protocol to the English source corpora. This involved systematically filtering out noisy, incomplete, or low-quality source rows and purging extensive cross-dataset duplicates. This preprocessing step ensured that only high-quality, medically relevant claims were advanced to the translation pipeline. Detailed statistics, including the breakdown of *real* and *fake* claims for each sub-corpus, are presented in Table 2. This resource was specifically curated to evaluate how modern LLMs handle the semantic nuances of sensitive medical claims when transitioned from high-resource English to the low-resource South Asian context of the Nepali language.

The construction of Nep-Health-Misinfo followed a multi-stage pipeline designed to maximize linguistic and medical accuracy. In the initial phase, we adhered to specific preservation protocols where @mentions, #hashtags, abbreviations (e.g., WHO, CDC), and specific identifiers like COVID-

Table 2: Distribution of real and fake health claims across the four sub-corpora of the Nep-Health-Misinfo dataset.

Dataset	Label	# Samples
CoAID	Real	441
	Fake	25
COVID-19	Real	1,050
	Fake	930
Monkeypox-V1	Real	1,329
	Fake	903
Monkeypox-V2	Real	158
	Fake	88
Total		**4,924**

19 were retained in their original English format during translation. We then established an automated baseline by processing the source text through three SOTA NMT engines: Google NMT, NLLB-200 (600M), and mBART-50. These models provided a diverse set of initial hypotheses, allowing us to identify consistent failure points in Devanagari script generation, specifically regarding the transliteration of Western biomedical terms versus native Nepali nomenclature.

To elevate the dataset to a gold-standard level, we conducted a rigorous MTPE phase on the entire corpus. Two native Nepali speakers were recruited for a careful sentence-by-sentence manual review. The editors were tasked with three primary objectives: ensuring strict medical fidelity, correcting grammatical errors and mistranslations introduced by NMT, and adapting Western-centric misinformation narratives into natural, culturally relevant Nepali phrasing without altering the ground truth. The original labels from the source datasets were preserved throughout translation and post-editing, and annotators did not modify class assignments. Disagreements between annotators were resolved through discussion, and unresolved cases were adjudicated by a senior reviewer.

Following the MTPE process, a final verification pass was conducted to ensure absolute uniqueness and linguistic consistency. Additionally, we anonymized specific user mentions to @MENTION to ensure models focus on the informational content rather than exhibiting bias toward specific entities; however, #hashtags and abbreviations were preserved in their original form.

Finally, we quantified the necessity of this human-verified data by evaluating the raw NMT outputs against our final human-edited version using BLEU, chrF, and Translation Edit Rate (TER) metrics. As summarized in Table 3, while Google NMT provided the strongest baseline, it achieved a TER of 40.95 on the CoAID dataset, indicating a high level of edit operations required during post-editing, reflecting substantial translation effort. Furthermore, our label-specific analysis revealed a complexity gap: misinformation instances *fake* yielded a significantly higher TER (62.42) compared to factual news (40.95). This suggests that health misinformation relies on linguistic irregularities that defy automated translation, thereby necessitating rigorous MTPE used in this study.

Table 3: Quantitative evaluation of translation models across datasets, stratified by veracity. Bold values highlight the top model per metric for each label, while red text indicates the global best for that metric across the specific dataset.

Dataset	Label	Metric	Google	NLLB	mBART
CoAID	Real	BLEU	**43.21**	28.44	17.39
		chrF	**71.76**	61.84	51.53
		TER ↓	**40.95**	57.33	67.04
	Fake	BLEU	**19.11**	12.56	2.14
		chrF	**65.92**	58.82	36.58
		TER ↓	**62.42**	78.34	85.99
COVID-19	Real	BLEU	**32.16**	23.57	15.62
		chrF	**66.91**	57.50	51.27
		TER ↓	**48.72**	62.44	72.10
	Fake	BLEU	**34.05**	22.99	15.75
		chrF	**68.85**	59.79	52.01
		TER ↓	**49.98**	68.82	74.70
Monkeypox-V1	Real	BLEU	**32.68**	21.94	16.87
		chrF	**64.35**	52.28	47.95
		TER ↓	**53.76**	73.70	76.25
	Fake	BLEU	**29.62**	21.66	13.89
		chrF	**60.08**	49.32	44.70
		TER ↓	**56.56**	70.93	79.81
Monkeypox-V2	Real	BLEU	**34.90**	19.02	15.04
		chrF	**65.39**	53.05	48.37
		TER ↓	**53.53**	71.82	80.85
	Fake	BLEU	**30.55**	22.26	14.51
		chrF	**56.73**	46.30	42.88
		TER ↓	**57.48**	70.35	78.74

3.2. Models and Experimental Settings

We benchmarked seven open-weight models selected for their parameter efficiency and multilingual capabilities: Qwen2.5-7B, Gemma-3-4B, Ministral-8B, Falcon3-7B, Phi-4-Mini, Mistral-7B, and Falcon-H1-3B. To rigorously evaluate in-context learning (ICL) capabilities in a low-resource setting, we tested each model under five distinct configurations: (i) zero-shot classification,

Example of Zero-shot prompt

Task: Health Misinformation Detection
Instruction: Determine if the following tweet in Nepali contains health misinformation.
Output 'Yes' if it is misinformation (fake) or 'No' if it is factual (real).
Output ONLY the word 'Yes' or 'No'.

Figure 1: Example of a Zero-shot prompt for Health Misinformation Detection.

Example of Few-shot prompt

Task: Health Misinformation Detection, **Instruction:** Determine if the following tweet in Nepali contains health misinformation. Output 'Yes' if it is misinformation (fake) or 'No' if it is factual (real). Output ONLY the word 'Yes' or 'No'.

P1	**P2**	**P3**	**P4**
Tweet: बिरालो र कुकुरले कोरोनाभाइरस फैलाउँछन् । **Translation:** Cats and dogs spread coronavirus.	**Tweet:** वैज्ञानिकहरू भन्छन् कि अमेरिका मंकीपक्स विरुद्धको लडाइँ हार्दै गइरहेको हुन सक्छ। **Translation:** Scientists say that America might be losing the fight against monkeypox.	**Tweet:** गेट्स फाउन्डेसनसँग यस कोरोनाभाइरसको प्याटेन्ट छ। **Translation:** The Gates Foundation has the patent for coronavirus.	**Tweet:** खैर यो मान्छेलाई पक्कै पनि समलिङ्गी भएको कारण मंकीपक्स भएको छ। **Translation:** Well, this guy definitely got monkeypox because he is gay.
Misinformation: Yes	**Misinformation:** No	**Misinformation:** Yes	**Misinformation:** Yes
Output: Yes	**Output:** No	**Output:** Yes	**Output:** Yes

Figure 2: Example of a Few-shot prompt for Health Misinformation Detection.

(ii) 5-shot random, (iii) 10-shot random, (iv) 5-shot sampled, and (v) 10-shot sampled. In the random setting, exemplars were selected uniformly at random from the training pool using a fixed seed.

Following Thapa et al. (2024), we used a clustering-based exemplar selection strategy for the sampled few-shot setting. For each shot size $k \in \{5, 10\}$, we constructed a label-balanced exemplar set by selecting approximately $k/2$ examples from each class *real* and *fake* whenever possible. Within each class subset, we represented Nepali texts using TF-IDF features with up to 500 dimensions and both unigram and bigram n-grams. We then applied K-means clustering with the number of clusters set to the required number of exemplars for that class. For each cluster, we selected the instance closest to the centroid as the exemplar. If clustering was not feasible due to insufficient data or feature sparsity, we fell back to random sampling within the class. The zero-shot prompt template used across experiments is illustrated in Figure 1, while the structure of the few-shot prompt with labeled exemplars is shown in Figure 2. All experiments were conducted using the vLLM library (Kwon et al., 2023) on L4 GPUs within the Google Colab ecosystem. To ensure reproducibility, random selection of few-shot exemplars was controlled with a fixed seed (seed = 1). For clustering-based sampled selection, K-means initialization was fixed internally (seed = 42), producing deterministic exemplar sets. This careful control of randomization ensures that all experiments can be exactly reproduced, allowing consistent comparison of model performance across different prompting settings and datasets. It also minimizes variability introduced by random exemplar selection while maintaining methodological rigor.

4. Results and Analysis

Our comprehensive benchmarking revealed distinct performance hierarchies among the evaluated architectures. The complete Macro F1 performance comparison across datasets and prompting configurations is presented in Table 4. In Table 4, random few-shot results were obtained using a fixed seed (seed = 1), while K-means-based sampled settings used a fixed initialization seed (seed = 42). Qwen2.5-7B was the most consistently strong overall performer across datasets. It achieved the highest peak Macro F1-scores on multiple sub-corpora, including 0.8488

Table 4: Performance comparison across datasets under different prompting settings. Bold text highlights the optimal model for each setting, while red text indicates the overall best model for each dataset.

Dataset	Setting	Qwen2.5-7B	Gemma-3-4B	Ministral-8B	Falcon3-7B	Phi-4-Mini	Mistral-7B	Falcon-H1-3B
Monkeypox-V1	Zero-Shot	**0.7188**	0.3255	0.3361	0.4922	0.4966	0.4170	0.5946
	5-Shot Sampled	**0.8029**	0.7032	0.4622	0.7659	0.7163	0.6187	0.7564
	10-Shot Sampled	**0.8121**	0.6945	0.6827	0.7475	0.7679	0.7012	0.7240
	5-Shot Random	0.6662	0.8181	0.8301	0.7647	0.7448	0.6764	0.8033
	10-Shot Random	**0.8235**	0.7505	0.7546	0.7552	0.7836	0.7566	0.7956
Monkeypox-V2	Zero-Shot	**0.7061**	0.2903	0.3179	0.4971	0.4582	0.3480	0.5908
	5-Shot Sampled	0.6849	0.6938	0.4359	0.6819	**0.6928**	0.5115	0.6534
	10-Shot Sampled	0.7283	0.6549	0.5731	**0.8039**	0.7130	0.5947	0.6690
	5-Shot Random	0.7918	0.7449	0.5913	**0.8028**	0.7945	0.6014	0.6530
	10-Shot Random	0.8488	0.7748	0.8449	0.8060	0.8003	0.6862	0.6970
CoAID	Zero-Shot	0.3413	0.0678	0.2254	0.1480	0.2325	0.1452	**0.6427**
	5-Shot Sampled	0.5771	0.4890	0.1723	0.6634	0.3643	0.6183	**0.7252**
	10-Shot Sampled	**0.7453**	0.5731	0.1995	0.5671	0.4132	0.5917	0.6303
	5-Shot Random	**0.7594**	0.3679	0.0606	0.5912	0.2108	0.4914	0.6693
	10-Shot Random	0.7798	0.6366	0.6220	0.6460	0.4812	0.6729	0.6290
COVID-19	Zero-Shot	**0.7153**	0.4231	0.4417	0.6153	0.6556	0.6414	0.5107
	5-Shot Sampled	0.7386	**0.8161**	0.5268	0.6025	0.7755	0.7782	0.6249
	10-Shot Sampled	0.7458	0.8368	0.8358	0.5488	0.7764	0.7822	0.5207
	5-Shot Random	0.6806	**0.7979**	0.7058	0.6443	0.7472	0.7451	0.6287
	10-Shot Random	0.7315	**0.8317**	0.8170	0.6865	0.7054	0.7324	0.7324

on Monkeypox-V2 and 0.7798 on CoAID in the 10-shot random setting. However, no single model dominated all configurations; models such as Gemma-3-4B and Ministral-8B achieved competitive or superior performance in specific datasets and prompting settings, indicating that model effectiveness varies depending on task characteristics and experimental setup. While Qwen2.5-7B demonstrates strong cross-dataset robustness, attributing this performance solely to pretraining factors (e.g., exposure to Asian linguistic data or Devanagari scripts) remains speculative and requires further controlled analysis.

Conversely, zero-shot performance was generally lower than few-shot settings across most datasets and models; for instance, Gemma-3-4B achieved a Macro F1 of only 0.068 on CoAID. This zero-shot gap underscores the necessity of in-context grounding for LRL, as models frequently fail to map Nepali medical terminology to the correct label without structured demonstrations. Interestingly, while K-means sampling was hypothesized to outperform random selection, our results show that 10-shot random prompting often yielded higher peak scores. This suggests that in the presence of the semantic noise inherent in misinformation, simple random sampling may provide sufficient lexical and contextual diversity, performing comparably to or even better than centroid-based exemplar selection strategies.

4.1. The Random Sampling Advantage

A critical and novel finding of this study is the divergence of LRL performance from established trends in English-centric benchmarks. While prior literature typically advocates for clustering-based sampling to ensure diverse semantic coverage, our results for Nepali indicate that 10-shot random selection frequently enables models to achieve higher performance peaks.

To validate this trend, we performed an explicit multi-seed evaluation for our leading models, Qwen2.5-7B and Falcon3-7B, across three distinct random seeds (1, 123, and 2026). On the Monkeypox-V2 dataset, Qwen2.5-7B achieved a peak Macro F1-score of 0.8488 (seed = 1), while Falcon3-7B reached 0.8060 (seed = 1). Both scores significantly exceeded their deterministic clustering-based baselines. This trend persisted on the larger COVID-19 dataset, where Qwen2.5-7B reached a peak of 0.7612 (seed = 2026), surpassing the 0.7458 achieved through K-means clustering.

While multi-seed evaluation shows that performance is sensitive to exemplar selection, results vary noticeably across random seeds. Despite this, random sampling often outperformed clustering-based sampling in our reported runs. We hypothesize that in LRL settings, clustering-based selection may limit the diversity of contextual signals provided to the model, whereas random sampling captures a broader range of linguistic patterns. This suggests that, in our setting, lexical and semantic diversity in exemplars may be more important than centroid-based selection.

4.2. Inter-Model Agreement

Assessing the consistency of predictions across diverse architectures is crucial for ensuring system reliability. We utilized Fleiss' Kappa (κ) to evalu-

Table 5: Fleiss Kappa Scores Across Different Training Settings. Red text indicates the best setting for each dataset.

Dataset	Setting	Fleiss' Kappa
COVID-19	10-Shot Random	**0.5115**
	10-Shot Sampled	0.4025
	5-Shot Random	0.3549
	5-Shot Sampled	0.2910
	Zero-Shot	0.0874
CoAID	10-Shot Random	**0.2988**
	10-Shot Sampled	0.1627
	5-Shot Random	0.0010
	5-Shot Sampled	0.1479
	Zero-Shot	0.0148
Monkeypox-V1	10-Shot Random	**0.4568**
	10-Shot Sampled	0.3667
	5-Shot Random	0.4204
	5-Shot Sampled	0.2798
	Zero-Shot	0.0036
Monkeypox-V2	10-Shot Random	**0.4764**
	10-Shot Sampled	0.2700
	5-Shot Random	0.2923
	5-Shot Sampled	0.1740
	Zero-Shot	0.0073

ate agreement among the seven evaluated models. The resulting inter-model agreement scores across datasets and prompting configurations are summarized in Table 5. Our analysis reveals that inter-model agreement increases with the number of in-context demonstrations provided.

In zero-shot settings, inter-model agreement was extremely low ($\kappa \approx 0.00$–0.08), indicating highly inconsistent predictions across models. However, in 10-shot settings, agreement rose to moderate levels ($\kappa > 0.51$ on COVID-19). This convergence suggests that providing sufficient exemplars encourages divergent architectures to adopt a more consistent decision framework for identifying Nepali misinformation, effectively aligning the models through the provided context.

4.3. Qualitative Analysis: Systemic Failure Modes

To deepen our understanding of model limitations, we conducted a granular error analysis, isolating instances where all seven models failed across both 10-shot settings ($n = 14$). This analysis revealed three systemic failure modes that characterize the current limitations of generative AI in the Nepali health domain.

First, models exhibited a pronounced Authority Bias. They consistently misclassified fabricated claims as Real when the text contained the names of high-prestige international or national institutions. For example, a fabricated claim stating "UNICEF releases coronavirus prevention guidelines" (ID 647) was universally accepted as factual. As shown in Table 6, this heuristic shortcut appears to override critical analysis; similar failures occurred with claims attributing false information to national medical bodies such as AIIMS (ID 1192).

Second, we observed a vulnerability to Pseudo-Scientific Mimicry. Misinformation utilizing sophisticated medical vocabulary was frequently labeled as factual. This was evident in instances where models failed to flag deceptive content containing terms like "genome" and "host immunity" (ID 868). This suggests that models rely on a complexity heuristic, where domain-specific vocabulary is treated as a proxy for truthfulness, failing to distinguish between genuine medical discourse and stylistic mimicry.

Finally, we observed a significant trend of Safety-Alignment Bias. Factual news reporting on LGBTQ+ health issues, specifically regarding Monkeypox, was frequently flagged as *fake*. As seen in Table 6 (ID 273 and 272), legitimate updates regarding vaccine availability and symptom reporting within the community were misclassified. This may indicate that alignment or safety tuning can sometimes make models overly cautious in cases involving minority identities and infectious disease.

5. Conclusion

This study introduces Nep-Health-Misinfo, the first human-verified Nepali health misinformation dataset. Through a comprehensive evaluation of seven recent open-weight models, we established that Qwen2.5-7B serves as a robust baseline for deployment in South Asian LRL settings. Our experimental results yield a significant methodological finding: stochastic (random) exemplar selection consistently reaches higher performance peaks than deterministic clustering-based strategies. This suggests that for Nepali health misinformation, capturing broad lexical diversity is more critical than maximizing semantic centroid representation. This finding simplifies the technical path for LRL deployment by reducing the computational overhead required for complex data preprocessing without compromising downstream accuracy.

However, our qualitative analysis revealed critical systemic vulnerabilities, specifically Author-

Table 6: Systemic Failure Analysis: Representative examples where all models failed ($n = 14$). The samples illustrate specific failure modes, including: Authority Bias (misleading reliance on prestige entities like UNICEF/AIIMS), Pseudo-Science (confusion caused by medical jargon), and Safety-Alignment Bias (misclassification of factual LGBTQ+ health news).

Dataset	Failure Mode	ID	Original Text	Translation	Truth	Pred
COVID-19	**Authority Bias**	647	UNICEF releases coronavirus prevention guidelines.	UNICEF ले कोरोनाभाइरस रोकथाम दिशानिर्देशहरू जारी गर्यो।	Fake	Real
		1192	AIIMS has released a list of respiratory symptoms of Covid-19 and similar diseases.	AIIMS ले Covid-19 र यस्तै रोगहरूको श्वासप्रश्वासका लक्षणहरूको सूची जारी गरेको छ।	Fake	Real
COVID-19	**Pseudo-Science**	868	there is variation in mortality and infection rate based upon the genome of the virus. Host immunity is also playing a role.	भाइरसको जीनोममा आधारित मृत्युदर र संक्रमण दरमा भिन्नता छ।होस्टको प्रतिरक्षाले पनि भूमिका खेलिरहेको छ।	Fake	Real
		216	The private health system began offering antibody tests to detect COVID-19. At the same time that the Ministry of Health said it would buy antigen tests.	निजी स्वास्थ्य प्रणालीले COVID-19 पत्ता लगाउन एन्टिबडी परीक्षणहरू प्रस्ताव गर्न थाल्यो। उही समयमा स्वास्थ्य मन्त्रालयले एन्टिजेन परीक्षणहरू खरिद गर्ने बतायो।	Fake	Real
Monkeypox-V1	**Safety Bias**	273	Pride Center Offers Free Vaccinations as Monkeypox Surges in Broward County	ब्रोवार्ड काउन्टीमा मङ्कीपक्स बढ्दै जाँदा प्राइड सेन्टरले निःशुल्क खोपहरू प्रदान गर्दछ।	Real	Fake
		272	I shot video interviews of gay men talking about their #monkeypox symptoms for my @MENTION article. They talk about the first signs of the infection and about severe pain. via @MENTION	मैले मेरो @MENTION लेखका लागि समलिङ्गी पुरुषहरूले आफ्ना #monkeypox लक्षणहरूबारे कुरा गरिरहेको भिडियो अन्तर्वार्ताहरू खिचें। उनीहरू सङ्क्रमणको पहिलो सङ्केत र गम्भीर पीडाको बारेमा कुरा गर्छन्। @MENTION मार्फत	Real	Fake

ity Bias and Safety-Alignment Bias. These findings serve as a necessary caution: current frontier models remain susceptible to deceptive authoritative tones and may inadvertently censor factual discourse concerning marginalized communities. By providing both a high-quality dataset and a quantified analysis of the current performance gap, this work establishes a foundational framework for bridging the digital language divide in public health safety. We anticipate that the Nep-Health-Misinfo corpus will catalyze future research into cross-lingual alignment and the development of safer, more equitable multilingual AI systems.

6. Limitations and Ethical Considerations

6.1. Linguistic and Data Limitations

While Nep-Health-Misinfo represents the first human-verified health misinformation dataset for Nepali health safety, several limitations persist. First, the dataset is derived from four English-centric benchmarks; consequently, it may not fully capture misinformation narratives that are indigenous to the South Asian socio-political landscape, such as specific Ayurvedic or local folk-medicine fallacies. Second, while our MTPE protocol utilized native experts to mitigate translation asymmetry, the source material's Western origin remains a latent bias in the dataset's thematic distribution. Future iterations should incorporate natively authored Nepali misinformation from local social media platforms to capture regional rhetorical styles.

6.2. Computational and Model Scope

Our benchmarking was restricted to seven open-weight models under 10 billion parameters to ensure feasibility for low-resource deployment. While Qwen2.5-7B emerged as a consistent model, the performance of significantly larger frontier models remains unmapped in this specific context. Furthermore, our study focused on text-only misinformation; however, real-world health deception in Nepal often propagates through multimodal formats, including deepfake audio and manipulated imagery, which fall outside the current scope.

6.3. Ethical Considerations

The curation and release of misinformation datasets necessitate strict ethical oversight. To prevent the accidental propagation of harmful claims, the Nep-Health-Misinfo corpus is intended strictly for research purposes. All instances in the dataset have been clearly labeled to prevent confusion with factual medical advice. Furthermore, during the construction phase, we performed rigorous anonymization of all user-specific metadata, replacing handles with `@MENTION`, to protect the privacy of the original posters in accordance with global data protection standards.

Finally, our discovery of safety-alignment bias regarding LGBTQ+ health reporting highlights a significant ethical risk: the potential for AI systems to inadvertently marginalize vulnerable populations under the guise of safety. We advocate for the responsible use of this dataset to audit and de-bias LLMs, ensuring that public health safety initiatives do not result in the systemic censorship of factual, life-saving information for minority communities.

References

U Acharya. 2023. Promoting digital literacy with scarce resources. *Managing the Misinformation Effect—The State of Fact-Checking in Asia. International Federation of Journalists*.

Daniel Alexander Alber, Zihao Yang, Anton Alyakin, Eunice Yang, Sumedha Rai, Aly A Valliani, Jeff Zhang, Gabriel R Rosenbaum, Ashley K Amend-Thomas, David B Kurland, et al. 2025. Medical large language models are vulnerable to data-poisoning attacks. *Nature Medicine*, 31(2):618–626.

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. 2025. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*.

Sirisha Bojjireddy, Soon Ae Chun, and James Geller. 2021. Machine learning approach to detect fake news, misinformation in covid-19 pandemic. In *Proceedings of the 22nd Annual International Conference on Digital Government Research*, pages 575–578.

Matteo Cinelli, Walter Quattrociocchi, Alessandro Galeazzi, Carlo Michele Valensise, Emanuele Brugnoli, Ana Lucia Schmidt, Paola Zola, Fabiana Zollo, and Antonio Scala. 2020. The covid-19 social media infodemic. *Scientific reports*, 10(1):16598.

Stephen Crone. 2022. Monkeypox misinformation: Twitter dataset.

Limeng Cui and Dongwon Lee. 2020. Coaid: Covid-19 healthcare misinformation dataset. *arXiv preprint arXiv:2006.00885*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Gunther Eysenbach et al. 2008. Medicine 2.0: social networking, collaboration, participation, apomediation, and openness. *Journal of medical Internet research*, 10(3):e1030.

Jay Gala, Pranjal A. Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M. Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages.

Plan Ghimire and Pranjal Shrestha. 2025. Bilingual fake-news detection in low-resource media: A transformer-based framework for nepali–english content. *Journal of Innovations in Engineering Education*, 8(1):133–138.

Saroj Giri, Shiva Ram Dam, Rajesh Kamar, and Suraj Basant Tulachan. 2024. Fake news detection using convolutional neural networks. *Technical Journal*, 4(1):64–68.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.

Anusha Kamath, Kanishk Singla, Rakesh Paul, Raviraj Bhuminand Joshi, Utkarsh Vaidya, Sanjay Singh Chauhan, and Niranjan Wartikar. 2025. Benchmarking hindi llms: A new suite of datasets and a comparative analysis. In *Proceedings of the 1st Workshop on Benchmarks, Harmonization, Annotation, and Standardization for Human-Centric AI in Indian Languages (BHASHA 2025)*, pages 52–68.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ayush Maheshwari, Kaushal Sharma, Vivek Patel, and Aditya Maheshwari. 2025. Indicparam: Benchmark to evaluate llms on low-resource indic languages. *arXiv preprint arXiv:2512.00333*.

Miriam J Metzger. 2007. Making sense of credibility on the web: Models for evaluating online

information and recommendations for future research. *Journal of the American society for information science and technology*, 58(13):2078–2091.

Parth Patwa, Shivam Sharma, Srinivas Pykl, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. 2021. Fighting an infodemic: Covid-19 fake news dataset. In *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*, pages 21–29. Springer.

Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, and Partha Talukdar. 2024. Indicgenbench: A multilingual benchmark to evaluate generation capabilities of llms on indic languages. *arXiv preprint arXiv:2404.16816*.

Briony Swire-Thompson and David Lazer. 2020. Public health and online misinformation: challenges and recommendations. *Annual review of public health*, 41:433–451.

Surendrabikram Thapa, Kritesh Rauniyar, Hariram Veeramani, Aditya Shah, Imran Razzak, and Usman Naseem. 2024. Did you tell a deadly lie? evaluating large language models for health misinformation identification. In *International Conference on Web Information Systems Engineering*, pages 391–405. Springer.

Emily K Vraga and Leticia Bode. 2020. Defining misinformation and understanding its bounded nature: Using expertise and evidence for describing misinformation. *Political Communication*, 37(1):136–144.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

John Zarocostas. 2020. How to fight an infodemic. *The lancet*, 395(10225):676.

Zhihao Zhang, Yiran Zhang, Xiyue Zhou, Liting Huang, Imran Razzak, Preslav Nakov, and Usman Naseem. 2025. From generation to detection: A multimodal multi-task dataset for benchmarking health misinformation. *arXiv preprint arXiv:2505.18685*.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–20.

Multimodal Hate and Sentiment Understanding in Low-Resource Text-Embedded Images for Online Safety and Digital Well-being

Surendrabikram Thapa[1], Shuvam Shiwakoti[1, †], Siddhant Bikram Shah[2, ✪], Kritesh Rauniyar[3, §], Laxmi Thapa[4], Surabhi Adhikari[5], Kristina T. Johnson[2], Kengatharaiyer Sarveswaran[6], Bal Krishna Bal[7], and Usman Naseem[3]

[1]Virginia Tech, USA, [2]Northeastern University, USA, [3]Macquarie University, Australia,
[4]O.P. Jindal Global University, India, [5]Columbia University, USA
[6]University of Jaffna, Sri Lanka, [7]Kathmandu University, Nepal
[†]shuvam@vt.edu; [✪]shah.siddhantb@northeastern.edu; [§]rauniyark11@gmail.com

Abstract

This paper presents an overview of the Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes, organized as part of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) at LREC 2026. The task addresses automated content understanding in low-resource settings by focusing on monolingual Nepali memes written in Devanagari script. Built upon the NeMeme dataset, the task comprises two subtasks: (1) binary hate speech detection and (2) three-class sentiment analysis. The competition attracted 23 teams for hate detection and 13 teams for sentiment analysis. Participating teams employed diverse strategies, including late-fusion multimodal architectures combining multilingual text encoders with vision models, caption-based approaches using large vision-language models, and ensemble techniques. The top-performing system achieved macro-F1 scores of 80.52% on hate detection and 68.81% on sentiment analysis using a late-fusion hybrid architecture with discriminative learning rates. Our analysis reveals that multimodal fusion consistently outperforms unimodal baselines, sentiment analysis poses greater challenges than hate detection due to increased semantic nuance, and the scarcity of Devanagari-centric pretrained models remains a significant bottleneck. This shared task establishes a benchmark for multimodal understanding in low-resource South Asian languages and provides insights for developing inclusive content moderation systems.

Keywords: Hate Speech Detection, Sentiment Analysis, Low-Resource Languages, Nepali Language, Vision-Language Models, Devanagari Script

1. Introduction

The rapid proliferation of social media has fundamentally transformed how individuals express opinions, disseminate ideas, and engage in public discourse across linguistic and cultural boundaries (Bhandari et al., 2023; Shiwakoti et al., 2024). Among the diverse forms of digital communication, memes–multimodal artifacts that interweave images and embedded text–have emerged as one of the most pervasive and influential vehicles for conveying humor, political commentary, social criticism, and cultural identity (Shifman, 2013). Their virality and accessibility make them powerful instruments for shaping public sentiment, but they also serve as conduits for hate speech, misinformation, and targeted abuse against individuals and communities (Pramanick et al., 2021). The inherent ambiguity of memes, where the boundary between satire and genuine malice is often deliberately blurred, presents a formidable challenge for both human moderators and automated content moderation systems (Shah et al., 2024; Shang et al., 2021).

While significant progress has been made in the computational analysis of memes in high-resource languages such as English (Kiela et al., 2020), the overwhelming majority of the world's languages remain critically underserved by existing tools and resources. This disparity is particularly consequential for languages that are widely spoken yet lack the annotated datasets, pretrained models, and benchmark evaluations necessary to drive progress in multimodal understanding. Nepali, a Devanagari-script language spoken by over 20 million people worldwide and the official language of Nepal, exemplifies this challenge. Despite the strong presence of politically and socially driven memes in Nepali online spaces, computational resources for understanding such content remain scarce (Thapa et al., 2025a). The low-resource nature of Nepali, compounded by its rich morphological structure, cultural specificity, and the prevalence of code-mixed expressions in digital communication, renders the direct application of models developed for high-resource languages both unreliable and culturally inadequate.

The challenge of analyzing Nepali memes is amplified by several interrelated factors. First, the multimodal nature of memes requires systems to jointly reason over visual and textual modalities–a task complicated by the fact that dominant vision-language models such as CLIP (Radford et al., 2021) are predominantly pretrained on English-centric web data and exhibit limited representa-

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 219–228
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

tional capacity for Devanagari script and culturally specific South Asian visual content (Poudel et al., 2025). Second, the nuanced interplay between humor, sarcasm, political satire, and genuine hate in Nepali memes demands models that can disentangle layered communicative intents rather than relying on surface-level lexical or visual cues (Rai et al., 2023). Third, the scarcity of annotated data imposes severe constraints on model training, necessitating innovative strategies for data augmentation, cross-lingual transfer, and efficient fusion of multimodal signals (Thapa et al., 2025a).

To address these critical research gaps, we present the **Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes**, organized as part of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) (Sarveswaran et al., 2026), co-located with LREC 2026. Building on the NeMeme dataset (Thapa et al., 2025a)–the first publicly available annotated dataset for hate speech and sentiment analysis in Nepali memes–this shared task is designed to catalyze the development of robust multimodal systems capable of operating in low-resource settings. The task encompasses two complementary subtasks: (1) **Hate Speech Detection**, a binary classification task requiring systems to distinguish hateful memes from non-hateful ones, and (2) **Sentiment Analysis**, a three-class classification task requiring systems to assign negative, neutral, or positive sentiment labels to memes. Both subtasks operate exclusively on monolingual Nepali memes written in the Devanagari script, evaluated using macro-averaged F1-score to account for inherent class imbalance.

The shared task attracted participation from a diverse set of teams employing a wide spectrum of methodological strategies, ranging from unimodal text-only and vision-only baselines to multimodal fusion architectures incorporating cross-modal attention, learnable gating mechanisms, and vision-language model-generated captions. Participants explored innovative approaches to circumvent the limitations of existing pretrained models for Devanagari content, including the use of large vision-language models such as Gemini for stochastic caption generation, multilingual encoders such as BGE-M3 and XLM-RoBERTa for text representation, and various fusion strategies, including early concatenation, late voting, weighted fusion, and hybrid attention-based architectures. The diversity of submitted approaches and their varying degrees of success provide valuable insights into the relative merits of different multimodal strategies under extreme data scarcity and linguistic resource constraints.

This paper provides a comprehensive overview of the shared task, including a description of the dataset and its annotation, the evaluation protocol and metrics for each subtask, a summary of the participating teams and their methodologies, and an analysis of the results. Through this shared task, we aim to advance the state of multimodal content understanding for low-resource South Asian languages and contribute to the development of more inclusive, culturally aware, and effective systems for online safety and digital well-being.

2. Related Works

Detecting hate speech on social media has become an increasingly important area of research, with most efforts concentrating on text-based content (Chhabra and Vishwakarma, 2023; Parihar et al., 2021). However, there has been a notable surge in scholarly interest in identifying hate speech in memes or text-containing images (Ji et al., 2023; Thapa et al., 2025b), as multimodal hate and sentiment analysis on social media remains an increasingly complex challenge for researchers, policymakers, and society (Jahan and Oussalah, 2023). Text-embedded images, including memes and other visual-text content like TV headline screenshots, combine images for context and text for information. While meme analysis has received significant attention, detecting hate speech in such text-embedded images remains underexplored, motivating this shared task.

Similarly, research on memes and multimodal textual-visual data has predominantly focused on general social media platforms, with an emphasis on the English language (Hossain et al., 2022; Xu et al., 2022; Shah et al., 2024). Efforts to create dedicated datasets for low-resource languages are quite limited, especially for the Nepali language, which is spoken by approximately 40 million people worldwide (Thapa et al., 2023; Poudel et al., 2025). A few studies have explored the Hindi language, where researchers curated multimodal datasets of memes and sarcastic content (Dubey et al., 2025; Kumari et al., 2024). However, in the context of the Nepali language and multimodal memes, there is a severe lack of datasets that comprehensively capture multiple aspects of social dynamics, such as cultural nuances, humor, sentiment, and hate speech, limiting research in this underexplored domain. Thapa et al. (2025a) introduced a unique dataset featuring multimodal Nepali and code-mixed Nepali memes (combining Nepali and English). These efforts are progressively laying the foundation for future research on language-specific multimodal content. This shared task aims to engage the research community and encourage their participation in context-specific investigations, specifically focusing on Nepali meme identification.

3. Shared Task Description

Our shared task targets the automated understanding of monolingual Nepali memes across two complementary subtasks: hate speech detection and sentiment analysis. Both subtasks operate exclusively on memes written in the Nepali language using the Devanagari script, posing unique challenges due to the low-resource nature of the language, its rich cultural context, and the inherently multimodal nature of meme content.

3.1. Subtask A: Hate Speech Detection

Hate Speech Detection in Nepali Memes: The aim of this subtask is to detect the presence of hate speech in monolingual Nepali memes. Given a meme, participating systems must classify it into one of two categories: *Non-Hate* and *Hate*. Hateful memes are defined as those that vilify, denigrate, bully, insult, or mock a subject based on characteristics such as gender, race, religion, caste, or organizational status, as well as memes that contain hateful symbols or glorify violence. Non-hateful memes, by contrast, may convey humor, affection, motivation, or constructive criticism without malicious intent. This subtask is evaluated using macro-averaged F1-score to account for the natural class imbalance between hate and non-hate instances in the dataset.

3.2. Subtask B: Sentiment Analysis

Sentiment Analysis in Nepali Memes: The goal of this subtask is to classify the sentiment expressed in monolingual Nepali memes. Given a meme, participating systems must assign one of three sentiment labels: *Negative*, *Neutral*, or *Positive*. Negative memes are those that aim to denigrate, insult, or belittle a subject based on their social, personal, or organizational status. Neutral memes present an objective perspective without conveying strong emotional tones. Positive memes express affection, support, gratitude, praise, or motivation. This subtask is evaluated using macro-averaged F1-score across all three sentiment classes.

4. Dataset

This section describes the dataset used in our shared task, including its source, annotation procedure, and the statistics of the train, validation, and test splits used for evaluation.

4.1. Source and Collection

We utilize a subset of the NeMeme dataset (Thapa et al., 2025a), the first publicly available annotated dataset for hate speech and sentiment analysis in Nepali memes. The full NeMeme dataset was collected from multiple social media platforms, including Twitter, Instagram, Reddit, Facebook, and Threads. For our shared task, we focus exclusively on the monolingual Nepali subset, which consists of memes written entirely in the Nepali language using the Devanagari script. Code-mixed memes combining Nepali and English were excluded from this task.

4.2. Annotation

Each meme in the dataset is annotated for two tasks: (1) binary hate speech detection (hate vs. non-hate) and (2) three-class sentiment analysis (negative, neutral, positive). Annotations were produced by three annotators proficient in Nepali, Hindi, and English, following a three-phase annotation scheme (Bhandari et al., 2023) comprising a pilot run, a revision phase, and a consolidation phase. Inter-annotator agreement, measured using Fleiss' κ, was 0.74 for hate speech and 0.69 for sentiment, indicating substantial agreement.

4.3. Data Splits

We partition the dataset into training, validation, and test sets for each task. Table 1 summarizes the label distributions across splits.

For the **hate speech detection** task, the training set contains 1,068 memes (348 hate, 720 non-hate), the validation set contains 133 memes (35 hate, 98 non-hate), and the test set contains 134 memes (44 hate, 90 non-hate). The dataset exhibits a natural class imbalance, with non-hateful memes comprising approximately 67% of instances across all splits, reflecting realistic distributions on social media platforms.

For the **sentiment analysis** task, the training set contains 1,061 memes (341 negative, 473 neutral, 247 positive), the validation set contains 133 memes (39 negative, 65 neutral, 29 positive), and the test set contains 133 memes (40 negative, 63 neutral, 30 positive). Neutral memes represent the plurality class, consistent with the broader NeMeme dataset statistics.

Table 1: Dataset statistics for the shared task across train, validation, and test splits.

Task	Label	Train	Val	Test	Total
Hate Speech	Hate	348	35	44	427
	Non-Hate	720	98	90	908
	Total	1068	133	134	1335
Sentiment	Negative	341	39	40	420
	Neutral	473	65	63	601
	Positive	247	29	30	306
	Total	1061	133	133	1327

Rank	Participant	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
1	linus (Regmi et al., 2026)	82.09	82.02	79.74	80.52
2	ZeroR (Khanal, 2026)	82.09	79.70	79.70	79.70
3	eGrantha.ai (Thapaliya, 2026)	78.36	72.85	76.12	73.97
4	samirwagle (Wagle et al., 2026)	70.15	70.23	68.17	68.34
5	team Oryu (Orny et al., 2026)	73.13	65.48	69.65	66.40
6	EthosAI (Bansal et al., 2026)	71.64	65.53	67.43	66.14
7	ashweta (Fondekar et al., 2026)	71.64	64.95	67.41	65.64
8	anish (Acharya et al., 2026)	70.90	64.97	66.52	65.50
9	CUET-DoubleA	65.67	66.89	64.93	64.33
10	sundess	67.16	62.78	62.78	62.78
11	sameekxa	64.93	63.43	62.24	62.36
12	hasnat (Chowdhury et al., 2026)	68.66	61.57	63.47	62.02
13	researcher	67.91	61.01	62.58	61.41
14	sujata-gaihre	64.93	61.69	61.12	61.30
15	Multi-Modal-Minds (Shrestha et al., 2026)	61.19	59.49	58.61	58.52
16	deleted_user_52441	64.93	58.21	59.06	58.42
17	eshan	57.46	55.56	55.01	54.72
18	dutta_arka	58.21	49.14	48.96	48.57
19	The Argonauts	67.91	52.88	62.82	47.94
20	now_coder	64.18	48.36	41.54	40.99
21	akshayyy22	67.16	50.00	33.58	40.18
22	sujatagaihre	67.16	50.00	33.58	40.18
23	anycookie	38.81	46.31	45.86	38.59

Table 2: Sub-task A (Hate Speech Detection) Leaderboard, Ranked by Macro F1-score. All scores are presented as percentages (%). Note that this leaderboard contains the score till the test deadline and does not consider further runs done by participants as a part of the system description paper.

Rank	Participant	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
1	linus (Regmi et al., 2026)	68.42	70.13	69.79	68.81
2	anish (Acharya et al., 2026)	56.39	54.52	56.59	55.18
3	sundess	71.43	60.91	50.67	54.81
4	ZeroR (Khanal, 2026)	52.63	53.57	52.68	51.77
5	samirwagle (Wagle et al., 2026)	60.15	53.48	55.23	51.12
6	sidratul (Muntaha et al., 2026)	51.13	52.01	50.70	50.45
7	dipika_nath (Debnath et al., 2026)	49.62	51.51	49.76	49.75
8	ashweta (Fondekar et al., 2026)	48.87	50.37	48.58	48.59
9	EthosAI (Bansal et al., 2026)	49.62	48.57	48.24	48.39
10	yinloonkhorr	49.62	47.10	47.89	47.26
11	researcher	44.36	40.77	42.22	41.08
12	Multi-Modal-Minds (Shrestha et al., 2026)	39.10	37.14	34.93	35.28
13	akshayyy22	45.11	36.59	36.85	33.36

Table 3: Sub-task B (Sentiment Analysis) Leaderboard, Ranked by Macro F1-score. All scores are presented as percentages (%). Note that this leaderboard contains the score till the test deadline and does not consider further runs done by participants as a part of the system description paper.

5. Evaluation and Competition

This section describes the structure of our competition, along with the methodology used to determine ranks and other relevant details.

5.1. Evaluation Metrics

To evaluate the effectiveness of the participants' contributions, we used four metrics: macro F1-score, accuracy, precision, and recall. The participants' final ranks were determined using the macro F1-score as the primary ranking metric.

5.2. Competition Setup

We used Codabench[1] [,2] to organize our competition. The competition was hosted as two separate competitions corresponding to the two subtasks. Each competition consisted of two phases: a development phase, where participants could familiarize themselves with the Codabench platform and de-

[1]Subtask A (Hate Detection): https://www.codabench.org/competitions/12090/

[2]Subtask B (Sentiment Analysis): https://www.codabench.org/competitions/12091/

velop their methods, and a test phase, where performance was used to determine the final ranking on the leaderboard. The results from the development phase were made available to participants after the phase concluded, enabling them to further refine their approaches for the test phase.

5.2.1. Registration

For Subtask A (Hate Detection), a total of 65 participants registered, out of which 23 teams submitted their predictions. For Subtask B (Sentiment Analysis), 43 participants registered, with 13 teams submitting their predictions.

5.2.2. Competition Timelines

The competition commenced on December 8, 2025, when training and development data were made available, marking the start of the development phase. During this phase, participants familiarized themselves with the Codabench platform and began developing their systems. The test phase began on December 25, 2025, when test data was provided without any ground truth labels. The test phase was originally scheduled to end on February 14, 2026; however, in response to requests from several participants, it was extended until February 22, 2026. The paper submission deadline was correspondingly updated from February 20, 2026 to March 1, 2026. Notification of acceptance was scheduled for March 20, 2026, with camera-ready papers due by March 30, 2026.

6. Participants' Methods

6.1. Overview

The leaderboards for Subtask A and Subtask B are presented in Table 2 and Table 3, respectively.

6.2. Methods

In the following subsections, we outline the approaches adopted by participating teams across both Subtasks A and B, as well as those specific to Subtask A and Subtask B individually, based on their system description papers.

6.2.1. Both Subtasks A and B

Linus (Regmi et al., 2026) proposed a late-fusion hybrid architecture combining CLIP ViT-B/32 for visual feature extraction with NepBERTa for textual encoding, targeting both Subtasks A and B. The extracted modality-specific embeddings are independently L2-normalised, concatenated into a 1280-dimensional joint representation, and passed through a multi-layer perceptron (MLP). Training employed discriminative learning rates, with substantially lower rates to prevent catastrophic forgetting. The training also used label smoothing, explicit class weighting for the underrepresented neutral class, and OneCycleLR scheduling. Their framework achieved Macro-F1 scores of 0.8052 and 0.6881 on Subtasks A and B, respectively, securing first position on both subtasks.

ZeroR (Khanal, 2026) introduced a two-stage vision-language system built on Qwen3-VL-8B-Instruct, leveraging the model's native Devanagari reading capability to process. In the first stage, LoRA adapters are applied across all linear projections for generative fine-tuning, accompanied by an MLP projection head mapping hidden states to a 512-dimensional embedding space. The second stage introduces supervised contrastive learning via InfoNCE loss jointly optimized with a linear classifier. The sentiment task demanded considerably stronger regularization, including lower LoRA rank, higher weight decay, and early stopping. The system achieved Macro-F1 scores of 0.7970 and 0.5177, securing second and fourth positions on Subtasks A and B, respectively.

MEME-Fusion (Wagle et al., 2026) proposed a hybrid cross-modal attention fusion architecture addressing both Subtasks A and B. The system combines CLIP ViT-B/32 for visual encoding with BGE-M3 for multilingual text representation, connected through 4-head self-attention and a learnable softmax-normalized gating network that dynamically weights modality contributions on a per-sample basis. Evaluation across eight model configurations demonstrated that the hybrid fusion model achieved a Macro-F1 of 0.6834 on Subtask A and 0.6102 on Subtask B, securing 4th and 5th place, respectively. The study also uncovered that English-centric vision models exhibited near-random performance on Devanagari script, and standard ensemble methods catastrophically degraded under data scarcity due to correlated overfitting.

EthosAI (Bansal et al., 2026) presented a multimodal approach for both Subtask A (Hate Speech Detection) and Subtask B (Sentiment Classification), systematically evaluating 10 image models and 10 text models before selecting the top two from each modality for fusion experiments. Text features were extracted using both EasyOCR and Qwen2-VL-7B-Instruct for image description generation, while four fusion strategies–simple concatenation, weighted fusion

with learnable scalars, cross-attention, and element-wise addition–were compared. For Subtask A, the best-performing configuration combined Sentence-Transformers/LaBSE for text and ResNet-18 for image using weighted fusion, achieving a Macro-F1 of 0.6614. For Subtask B, Sentence-Transformers/LaBSE with DeiT-Base using simple fusion achieved a Macro-F1 of 0.4839. Their experiments indicated that multimodal models consistently outperform unimodal baselines for meme understanding in low-resource settings.

Unigoa (Fondekar et al., 2026) investigated early-fusion versus late-fusion multimodal architectures for both Subtasks A and B. Their primary system employs a late-fusion dual-encoder design pairing XLM-RoBERTa with CLIP ViT-B/32, where text and image embeddings are concatenated and fed into an MLP classifier. For Subtask A, XLM-RoBERTa-Base is used alongside validation-based threshold tuning to handle class imbalance, while Subtask B adopts XLM-RoBERTa-Large with a suite of training stabilisation techniques including WeightedRandomSampler, Soft Focal Loss, gradient accumulation, and discriminative learning rates. As a comparative baseline, the authors also evaluated an early-fusion ViLT model with NepBERTa tokenization. The late-fusion approach consistently outperformed the early-fusion baseline, achieving Macro-F1 scores of 0.6564 and 0.4859 on Subtasks A and B, placing 7th and 8th respectively. Their analysis highlights the advantage of leveraging strong multilingual pretrained encoders in a late-fusion setup over joint vision–language transformers under low-resource data conditions, particularly for code-mixed memes containing both Devanagari and Roman-script text.

TeamHerald (Acharya et al., 2026) tackled both Subtasks A and B using a text-centric pipeline built on OCR-extracted text from meme images via EasyOCR, followed by fine-tuning six Transformer-based architectures including NepaliBERT, DistilBERT-Base-Nepali, XLM-RoBERTa-Base, Twitter-XLM-RoBERTa-Base, RoBERTa-Hindi, and a decoder-only distilgpt2-nepali model. Class imbalance was addressed through stratified oversampling prior to training. The authors systematically compared Hard Voting and Soft Voting ensemble strategies across both subtasks, revealing a task-dependent pattern: the standalone distilgpt2-nepali model achieved the best Macro-F1 of 0.6550 on Subtask A, while the Soft Voting ensemble yielded the highest Macro-F1 of 0.5518 on Subtask B, representing a 7.5% relative improvement over the best individual baseline. Their analysis suggests that probabilistic aggregation is more effective than majority voting for multi-class classification in low-resource settings, whereas binary tasks may benefit more from well-adapted standalone models.

Multi-Modal-Minds (Shrestha et al., 2026) conducted a broad comparative study spanning unimodal text models (mBERT, XLM-RoBERTa, MuRIL), unimodal visual models (ResNet, ConvNeXt, ViT), nine late-fusion combinations pairing each text encoder with each visual encoder, and the vision-language foundation model SigLIP. A stochastic image augmentation pipeline was used in all visual experiments to simulate web degradation, apply coarse dropout for spatial regularization, and introduce color variation. Surprisingly, instead of multimodal fusion performing best, the standalone ViT achieved the highest Macro-F1 on Subtask A (0.6278), while SigLIP performed best on Subtask B (0.5481). The architecture ranked 15th on Subtask A and 12th on Subtask B.

6.2.2. Subtask A

eGrantha.ai (Thapaliya, 2026) proposed a caption-based approach for hate speech detection using the Gemini family of models (Gemini 2.X and Gemini 3.X) to generate contextually rich captions from meme images, converting the multimodal problem into a standard text classification task. The system fine-tuned RoBERTa-base on the generated captions, employing a stochastic caption regeneration strategy to address class imbalance by doubling minority-class coverage. At inference time, Test-Time Augmentation (TTA) was applied using five independent captions generated by diverse Gemini model variants, with class-probability distributions averaged before argmax. This approach achieved a Macro-F1 of 0.7397, securing 3rd place on the official leaderboard, demonstrating that caption-based modeling with stochastic augmentation can compete with conventional multimodal fusion strategies.

team Oryu (Orny et al., 2026) only addressed Subtask A through a late-fusion multimodal framework pairing XLM-RoBERTa-Base for multilingual text encoding with a Vision Transformer (ViT-Base-Patch16-224) for image feature extraction. The 768-dimensional embeddings from each encoder are concatenated to form a 1536-dimensional joint representation. This representation is then processed by a two-layer fully connected classification head with ReLU activation and dropout regularization. The authors report an iterative development

process, progressing from a basic multimodal baseline (F1: 0.40) through a gated fusion variant (F1: 0.52) to the final late-fusion configuration. The system obtained a Macro-F1 of 66.40% on the official test set, ranking fifth on Subtask A.

HasNat (Chowdhury et al., 2026) tackled Subtask A by benchmarking three multimodal model families—ViT-B-32 (OpenCLIP), AltCLIP, and BLIP2+mT5—each evaluated under default, preprocessing, augmentation, and combined settings. All architectures employed a late-fusion strategy in which frozen vision–language backbones produced text and image embeddings that were concatenated and passed to a trainable MLP classifier. Training relied on 5-fold stratified cross-validation with weighted binary cross-entropy loss to handle class imbalance, and fold-level probability averaging for inference. Among the configurations explored, AltCLIP in its default setting yielded the highest validation Macro-F1 of 0.66, while the authors noted that preprocessing and augmentation effects were highly model-dependent, with aggressive augmentation sometimes degrading recall by distorting meme-embedded text.

6.2.3. Subtask B

NeuralNoodles (Muntaha et al., 2026) tackled Subtask B through a late-fusion multimodal stacking framework that integrates three complementary unimodal streams: a TF-IDF model combining word-level (1–2 gram) and character-level (3–5 gram) n-grams for lexical representation, a frozen `paraphrase-multilingual-MiniLM-L12-v2` Sentence Transformer for semantic encoding, and a fine-tuned EfficientNet-B0 for visual feature extraction. Each component produces a three-class probability distribution, which are concatenated into a 9-dimensional meta-feature vector and passed to a Logistic Regression meta-classifier trained via stratified 5-fold cross-validation. An ablation study revealed that while the image model was the strongest individual contributor (Macro-F1 of 0.3980), the full three-modality stack yielded a 6.4% relative improvement, achieving 0.4234 on cross-validation and 0.5045 on the official test set for 6th place. The authors note that two-modality combinations did not consistently outperform the best unimodal baseline, suggesting that complementary information across all three representation types is necessary for effective fusion in this low-resource setting.

Cuet Yet Another Baseline (Debnath et al., 2026) implemented a multimodal pipeline for sentiment Analysis that fuses three streams: contextual text features from a fine-tuned XLM-RoBERTa-large applied to OCR-extracted meme text combined with BLIP-2 captions, pooled visual features from a frozen CLIP ViT-L/14 encoder, and a BLIP-2 caption embedding along with a CLIP image–text cosine similarity scalar. The attended representation was concatenated with projected CLIP streams and passed through a two-layer MLP classifier with GELU activations and dropout. Code-mixed Nepali–English samples were incorporated as training augmentation, with monolingual samples upweighted via a WeightedRandomSampler. Their approach achieved a Macro-F1 of 0.50, securing 7th place.

7. Discussion

The results of this shared task reveal several important insights into multimodal hate speech and sentiment analysis for low-resource languages. The top-performing systems consistently demonstrated that late-fusion architectures pairing strong multilingual text encoders (NepBERTa, XLM-RoBERTa, BGE-M3) with established vision models (CLIP ViT) achieve superior performance compared to early-fusion or vision-language models like ViLT, suggesting that the scarcity of Devanagari-centric multimodal pretraining data makes separate encoding followed by learned fusion more effective than joint processing. Notably, the winning team's hybrid fusion with discriminative learning rates and careful regularization achieved macro-F1 scores of 80.52% and 68.81% on hate detection and sentiment analysis respectively, while caption-based approaches using large vision-language models like Gemini demonstrated competitive performance by converting the multimodal problem into text classification. The substantial performance gap between Subtask A (hate detection) and Subtask B (sentiment analysis), with the best F1 scores differing by approximately 12 percentage points, indicates that sentiment classification remains considerably more challenging, likely due to the inherent ambiguity in distinguishing neutral from mildly positive or negative memes and the greater semantic nuance required for three-way classification.

The diversity of methodological approaches adopted by participants underscores both the opportunities and challenges in low-resource multimodal understanding. While multimodal fusion generally outperformed unimodal baselines, the margin of improvement varied considerably across teams, with some fusion strategies yielding minimal gains or even degrading performance due to correlated overfitting under severe data scarcity. Vision-only models achieved surprisingly compet-

itive results on both subtasks, suggesting that meme images contain substantial visual cues independent of text, though text-centric models generally underperformed, highlighting the critical importance of visual context. The observed sensitivity to architectural choices, hyperparameter configurations, and augmentation strategies emphasizes the need for careful model design and extensive validation when working with low-resource languages. Future work should explore more sophisticated cross-lingual transfer learning, leverage larger multilingual vision-language models with improved Devanagari support, and investigate semi-supervised or self-supervised approaches to alleviate data scarcity constraints in Nepali and other underserved South Asian languages.

8. Conclusion

This paper presented an overview of the Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes, organized as part of CHiPSAL 2026 at LREC 2026. The results demonstrate that late-fusion architectures combining multilingual text encoders with established vision models achieve the most robust performance, with the top system attaining macro-F1 scores of 80.52% on hate detection and 68.81% on sentiment analysis. Key insights include the consistent superiority of multimodal fusion over unimodal approaches, the greater difficulty of sentiment classification compared to hate detection, and the competitive performance of caption-based methods using large vision-language models. We hope this shared task catalyzes further research into multimodal understanding for low-resource South Asian languages and contributes to the development of more inclusive, culturally aware content moderation systems for underserved linguistic communities.

9. References

Ashish Acharya, Anish Khatiwada, Rohit Khadka, and Pragya Aryal. 2026. Teamherald@chipsal 2026: Hate speech detection and sentiment analysis of nepali memes using transformer-based architectures and ensemble learning. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Vinayak Bansal, Deepawali Sharma, Aakash Singh, and Vivek Kumar Singh. 2026. Ethosai@chipsal2026: Hate and sentiment understanding in low-resource memes using a multimodal approach. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1994–2003.

Anusha Chhabra and Dinesh Kumar Vishwakarma. 2023. A literature survey on multimodal and multilingual automatic hate speech identification. *Multimedia Systems*, 29(3):1203–1230.

Alvee Hasan Chowdhury, MD. Abul Hasnat, and Adnan Faisal. 2026. Hasnat@chipsal 2026: Multimodal hate speech detection in low-resource nepali memes using aligned vision–language models. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Rotna Dipika Debnath, Shahrin Afroz Hoque Ruhi, Ayesha Labiba, Arpita Mallik, and Hasan Murad. 2026. Cuet yet another baseline@chipsal lrec 2026: Shared task on multimodal sentiment understanding in low-resource memes. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Kriti Dubey, Vaishnavi Srivastava, Garima Sharma, Nonita Sharma, Deepak Sharma, Uttam Ghosh, Osama Alfarraj, and Amr Tolba. 2025. Multimodal detection of offensive content in hindi memes. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Ashweta Fondekar, Milind Shivolkar, and Jyoti Pawar. 2026. Unigoa@chipsal 2026: Early vs late fusion for multimodal hate and sentiment detection in nepali memes. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. Mute: A multimodal dataset for detecting hateful memes. In *Proceedings of the 2nd conference of the asia-pacific chapter of the association for computational linguistics and the 12th international joint conference on natural language processing: student research workshop*, pages 32–39.

Md Saroar Jahan and Mourad Oussalah. 2023. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, page 126232.

Junhui Ji, Wei Ren, and Usman Naseem. 2023. Identifying creative harmful memes via prompt based approach. In *Proceedings of the ACM web conference 2023*, pages 3868–3872.

Nitiz Khanal. 2026. Zeror@chipsal 2026: Two-stage vision-language adaptation with contrastive learning for nepali meme classification. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624.

Gitanjali Kumari, Chandranath Adak, and Asif Ekbal. 2024. Mu2sts: a mu ltitask mu ltimodal s arcasm-humor-differential t eacher-s tudent model for sarcastic meme detection. In *European Conference on Information Retrieval*, pages 19–37. Springer.

Sidratul Muntaha, Sabila Anzum, Arpita Mallik, and Hasan Murad. 2026. Neuralnoodles@chipsal 2026: Late-fusion multimodal stacking for nepali meme sentiment classification. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Noore Tamanna Orny, Joyeta Barua Moni, and Md. Abtahee Kabir. 2026. Team oryu@chipsal 2026: Integrating text and vision transformers for multimodal hate speech detection in memes. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Sweta Poudel, Kritesh Rauniyar, Ashish Acharya, Junaid Rashid, Surabhi Adhikari, Usman Naseem, and Surendrabikram Thapa. 2025. Nepaes: exploring the promise of automated essay scoring for nepali essays. *PeerJ Computer Science*, 11:e3253.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. Momenta: A multimodal framework for detecting harmful memes and their targets. In *Findings of the association for computational linguistics: EMNLP 2021*, pages 4439–4455.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Pooja Rai, Sanjay Chatterji, and Byung-Gyu Kim. 2023. Deep learning-based sequence labeling tools for nepali. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(8):1–23.

Sunil Regmi, Bipesh Subedi, Saugat Singh, and Suman Shrestha. 2026. linus@chipsal 2026: Multimodal hate speech and sentiment detection in low-resource memes using late-fusion hybrid architecture. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Siddhant Bikram Shah, Shuvam Shiwakoti, Maheep Chaudhary, and Haohan Wang. 2024. Memeclip: Leveraging clip representations for multimodal meme classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17320–17332.

Lanyu Shang, Yang Zhang, Yuheng Zha, Yingxi Chen, Christina Youn, and Dong Wang. 2021. Aomd: An analogy-aware approach to offensive meme detection on social media. *Information Processing & Management*, 58(5):102664.

Limor Shifman. 2013. Memes in a digital world: Reconciling with a conceptual troublemaker. *Journal of computer-mediated communication*, 18(3):362–377.

Shuvam Shiwakoti, Surendrabikram Thapa, Kritesh Rauniyar, Akshyat Shah, Aashish Bhandari, and Usman Naseem. 2024. Analyzing the dynamics of climate change discourse on twitter: A new annotated corpus and multi-aspect classification. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pages 984–994.

Sandesh Shrestha, Bikram K.C., Akshyat Shah, Ashish Acharya, and Rabin Thapa. 2026. Multimodal-minds@chipsal 2026: A comparative

study of textual, visual and multimodal architecture for nepali meme moderation. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Kritesh Rauniyar, Shuvam Shiwakoti, Sweta Poudel, Usman Naseem, and Mehwish Nasim. 2023. Nehate: Large-scale annotated data shedding light on hate speech in nepali local election discourse. *Frontiers in Artificial Intelligence and Applications*, 372:2346–2353.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Anish Thapaliya. 2026. egrantha.ai@chipsal 2026: Stochastic image captioning for robust hate speech detection in low-resource nepali memes. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Samir Wagle, Reewaj Khanal, and Abiral Adhikari. 2026. Meme-fusion@chipsal 2026: Multimodal ablation study of hate detection and sentiment analysis on nepali memes. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Bo Xu, Tingting Li, Junzhe Zheng, Mehdi Naseriparsa, Zhehuan Zhao, Hongfei Lin, and Feng Xia. 2022. Met-meme: A multimodal meme dataset rich in metaphors. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2887–2899.

Unigoa@CHiPSAL 2026: Early vs Late Fusion for Multimodal Hate and Sentiment Detection in Nepali Memes

Ashweta A. Fondekar, Milind M. Shivolkar, Jyoti D. Pawar
Parvatibai Chowgule College of Arts and Science(Autonomous); Goa Business School, Goa University
aaf016@chowgules.ac.in, {dcst.ashweta, milind.shivolkar, jdp}@unigoa.ac.in

Abstract

Internet memes pose significant challenges for automatic content moderation due to the interaction of visual and textual cues, sarcasm, and cultural context. In this work, we participate in the CHiPSAL 2026 shared task on multimodal hate and sentiment understanding in Nepali memes. The task consists of two subtasks: binary hate speech detection and three-class sentiment classification. We investigate both early-fusion and late-fusion multimodal architectures. Our primary system employs a late-fusion dual-encoder architecture combining XLM-RoBERTa for multilingual text representation and CLIP for visual encoding. We further evaluate an early-fusion ViLT-based joint vision–language transformer using NepBERTa tokenization as a baseline. Experimental results show that late-fusion models consistently outperform early-fusion architectures, particularly for code-mixed memes containing Devanagari Nepali and Roman-script English text. Our best system achieves a Macro-F1 of 0.6564 for hate speech detection and 0.4859 for sentiment classification. We provide analysis highlighting the challenges of multilingual code-mixing, sarcasm, and implicit sentiment in low-resource multimodal settings.

Keywords: Multimodal Learning, Multimodal Fusion, Hate Speech Detection, Sentiment Analysis, Code-Mixed NLP, Nepali Memes

1. Introduction

Internet memes have emerged as a dominant form of communication on social media platforms. By combining images, short text, humor, and cultural references, memes enable users to express opinions and emotions in a compact and highly shareable format. However, the same characteristics that make memes engaging also make them difficult to moderate automatically. The meaning of a meme often depends on the interaction between visual content and text, and the intended message may be conveyed through sarcasm, irony, or implicit cultural context. Consequently, detecting hate speech or identifying sentiment in memes is considerably more challenging than in traditional text-only content (Parihar et al., 2021).

Despite recent progress in multimodal learning, most prior research has focused on English-language datasets. Low-resource languages such as Nepali remain underrepresented in multimodal content moderation research. Nepali social media discourse presents additional challenges due to informal language usage, frequent code-mixing between Nepali (Devanagari script) and English (Latin script), and culturally grounded humor. These characteristics significantly increase the difficulty of automatically identifying harmful or sentiment-bearing content.

This work participates in the CHiPSAL 2026 shared task on multimodal hate and sentiment understanding in Nepali memes (Thapa et al., 2026; Sarveswaran et al., 2026). The dataset builds upon previous research on multimodal meme understanding and code-mixed content (Thapa et al., 2025a,b) and follows annotation schemes inspired by CrisisHateMM (Bhandari et al., 2023). The shared task includes two subtasks:

- **Subtask A:** Binary hate speech detection.
- **Subtask B:** Three-class sentiment classification (positive, neutral, negative).

We investigate both early-fusion and late-fusion multimodal architectures. Our primary system uses a late-fusion dual-encoder architecture combining XLM-RoBERTa with CLIP. We additionally evaluate an early-fusion ViLT-based joint vision–language transformer using NepBERTa tokenization as a baseline. This design enables us to study the role of multilingual pretraining and fusion strategies in low-resource multimodal settings.

Our contributions are threefold:

- We propose a late-fusion multimodal architecture combining XLM-RoBERTa and CLIP for hate speech and sentiment detection in Nepali memes.
- We introduce a stabilised training strategy for sentiment classification using weighted sampling, soft focal loss, gradient accumulation, and discriminative learning rates.
- We provide an analysis of multilingual code-mixing and multimodal challenges in Nepali meme understanding.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 229–236
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

Paper Structure. The remainder of this paper is organised as follows. Section 2 reviews related work on multimodal hate speech and sentiment analysis. Section 3 describes the dataset and shared task. Section 4 presents the proposed multimodal architectures and training strategies. Section 5 details the experimental setup. Section 6 reports the results and analysis, and Section 7 concludes the paper.

2. Related Work

Early research on hate speech detection primarily focused on text-only analysis, highlighting linguistic challenges and societal implications (Parihar et al., 2021). With the rapid growth of image-centric social media, research has increasingly shifted toward multimodal analysis of text embedded in images.

The CrisisHateMM dataset introduced large-scale annotations for multimodal hate speech and demonstrated the importance of combining visual and textual signals (Bhandari et al., 2023). The Hateful Memes benchmark further emphasized the need for joint vision–language reasoning in meme understanding (Kiela et al., 2020). More recent work has explored multimodal meme understanding in code-mixed and low-resource settings using prompt-based and retrieval-augmented approaches (Thapa et al., 2025a,b).

Advances in multimodal representation learning have been driven by vision–language pretraining models such as CLIP (Radford et al., 2021) and ViLT (Kim et al., 2021), which learn shared representations across image and text modalities. Multimodal sentiment analysis has also been widely studied (Zadeh et al., 2018). In parallel, multilingual transformer models such as XLM-R (Conneau et al., 2020) have significantly improved modelling of code-mixed and low-resource languages.

The CHiPSAL shared task extends this line of research by focusing specifically on Nepali memes and low-resource multimodal moderation (Thapa et al., 2026; Sarveswaran et al., 2026). Our work builds upon these efforts by comparing early- and late-fusion multimodal architectures for Nepali meme understanding.

3. Dataset and Task Description

3.1. NeMeme Dataset Overview

We conduct experiments on the *NeMeme* dataset (Thapa et al., 2025a), a benchmark for multimodal meme understanding in Nepali. The dataset contains memes collected from social media platforms including Twitter, Reddit, Instagram, Facebook, and Threads. Each sample consists of an image paired with OCR-extracted text, enabling joint multimodal analysis of visual and textual information.

The memes are annotated by trained annotators for both hate speech and sentiment, and the dataset reports substantial inter-annotator agreement, indicating reliable labeling quality. The content reflects real-world Nepali social media discourse, including humor, sarcasm, informal language usage, and culturally grounded references.

Although the dataset is described as Nepali-only, manual inspection reveals frequent code-mixing between Nepali written in Devanagari script and English words written in Roman script.

3.2. Task Definition

The shared task consists of two classification problems:

- **Subtask A – Hate Speech Detection:** A binary classification task where memes are labeled as *hateful* or *non-hateful*.
- **Subtask B – Sentiment Classification:** A three-class classification task where memes are labeled as *negative*, *neutral*, or *positive*.

3.3. Linguistic Characteristics of the Dataset

Although the task is described as Nepali-only, the dataset reflects authentic Nepali social media usage, where English lexical insertions frequently appear within Nepali sentences. Manual inspection reveals substantial code-mixing between Nepali written in Devanagari script and English words written in Roman script.

Many memes contain English sentiment words, sarcasm markers, informal expressions, and references to meme pages embedded within Nepali discourse. In several cases, sentiment-bearing expressions appear in English, while the surrounding syntactic structure remains Nepali.

Such code-mixing mirrors real-world digital communication practices in Nepal and introduces additional challenges for monolingual models. The presence of mixed scripts, informal spelling variations, and culturally grounded humor increases the semantic complexity of both hate speech detection and sentiment classification. These linguistic characteristics strongly motivate the use of multilingual and multimodal encoders in our experiments.

3.4. Dataset Splits and Class Distribution

Tables 1 and 3 summarize the data splits for both subtasks, while Tables 2 and 4 present the label distributions in the training sets. The hate speech task exhibits noticeable class imbalance, which motivated the use of threshold tuning. For sentiment

Split (Hate Task)	Samples
Train	1068
Validation	133
Test	134

Table 1: Data split for hate speech detection.

Label (Hate Task)	Train Count
Hate (1)	720
Non-hate (0)	348

Table 2: Training label distribution for hate speech detection.

classification, class-weighted loss was applied to address imbalance across the three classes.

The validation sets follow a similar distribution. For hate speech detection, the validation set contains 98 hateful and 35 non-hateful memes. For sentiment classification, the validation set contains 39 negative, 65 neutral, and 29 positive memes.

4. Methodology

To address the two subtasks, we design multimodal systems that jointly model visual and textual information. While Subtask A focuses on binary hate detection, Subtask B involves three-way sentiment classification and presents additional challenges due to ambiguity, sarcasm, and code-mixed Nepali–English text. We therefore explore both early-fusion and late-fusion multimodal architectures to analyze the impact of fusion strategies and multilingual text encoders in low-resource meme understanding.

4.1. Subtask A: Multimodal Hate Speech Detection

We model hate speech detection as a binary multimodal classification problem. Each meme consists of an image and associated OCR text. Our primary system follows a late-fusion design with separate encoders for text and images.

Text is encoded using **XLM-RoBERTa-Base**, a multilingual transformer pretrained on large multilingual corpora. Token embeddings are aggregated using masked mean pooling to obtain a fixed-length sentence representation.

Images are encoded using **CLIP ViT-B/32**, which

Split (Sentiment Task)	Samples
Train	1061
Validation	133
Test	133

Table 3: Data split for sentiment classification.

Label (Sentiment Task)	Train Count
Negative (0)	341
Neutral (1)	473
Positive (2)	247

Table 4: Training label distribution for sentiment classification.

produces a pooled visual embedding. The text and image embeddings are concatenated and passed to a two-layer MLP classifier to produce binary logits.

Because the dataset is imbalanced, we apply **validation-based threshold tuning**. Instead of using the default decision threshold of 0.5, we select the probability threshold that maximizes Macro-F1 on the validation set before generating test predictions.

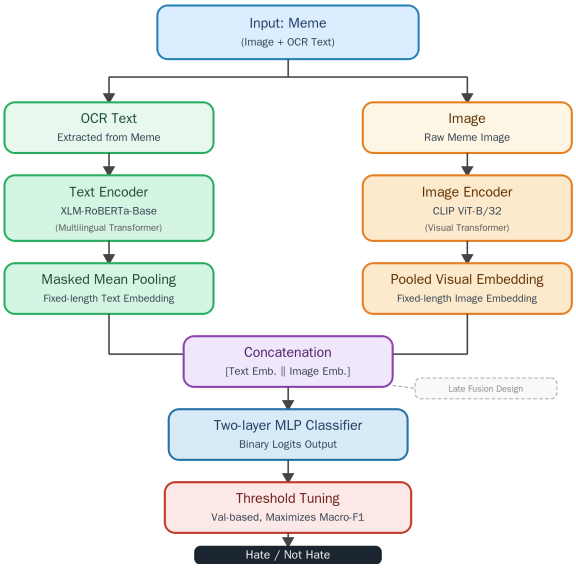


Figure 1: Late-fusion architecture for Subtask A (hate speech detection).

4.2. Subtask B: Multimodal Sentiment Classification

Sentiment classification is formulated as a three-class problem (negative, neutral, positive). We adopt a stronger variant of the fusion architecture using **XLM-RoBERTa-Large** for text encoding while retaining CLIP for image encoding.

Due to increased task difficulty and severe class imbalance, several training stabilisation techniques are introduced:

- WeightedRandomSampler for balanced mini-batches
- Soft Focal Loss with class-dependent weighting

- Gradient accumulation to increase effective batch size
- Discriminative learning rates for encoder vs classifier
- Freezing encoders during early training epochs

The fused multimodal embedding is passed to an MLP classifier with LayerNorm and GELU activations.

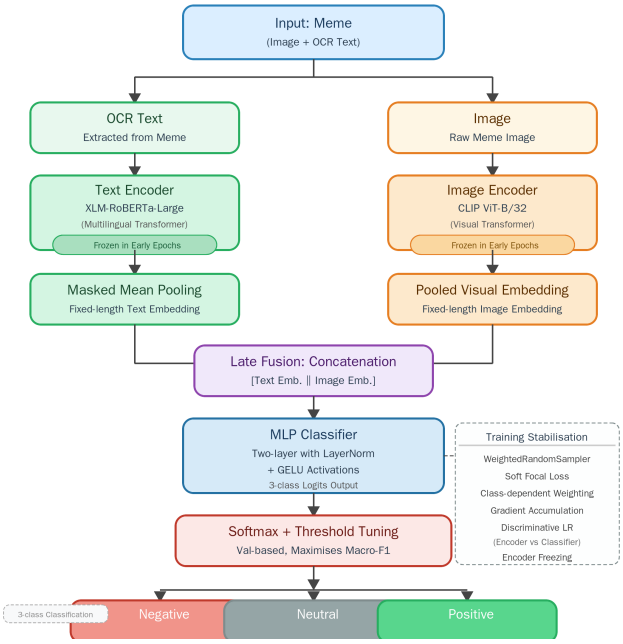


Figure 2: Late-fusion architecture for Subtask B (sentiment classification).

4.3. ViLT + NepBERTa (Early-Fusion Baseline)

To provide a comparative baseline, we evaluate a joint vision–language transformer using the **ViLT** architecture with **NepBERTa tokenization**. In this early-fusion approach, image patches and text tokens are processed jointly within a single transformer backbone, enabling cross-modal interactions at multiple layers.

The token embedding layer is resized to match the NepBERTa tokenizer vocabulary. The model is fine-tuned with task-specific classification heads for both subtasks using cross-entropy loss. This model serves as an early-fusion baseline against which our late-fusion XLM-R + CLIP models are compared. NepBERTa is selected for this model because it is specifically pretrained on Nepali text, allowing us to evaluate whether language-specific tokenization benefits early-fusion multimodal learning.

4.4. Model Comparison Strategy

We compare two multimodal fusion paradigms:

- **Early fusion:** ViLT + NepBERTa joint vision–language transformer.
- **Late fusion:** XLM-RoBERTa + CLIP dual-encoder architecture.

This comparison allows us to analyze how multimodal interaction strategies and multilingual pretraining influence performance on Nepali-dominant code-mixed memes. In particular, we investigate whether stronger multilingual text representations combined with late fusion can better capture sarcasm, humor, and culturally grounded expressions commonly found in Nepali memes. We intentionally restrict the comparison to two representative architectures to isolate the impact of multimodal fusion strategy under limited shared-task data. The ViLT + NepBERTa model represents a fully joint vision–language transformer trained in an early-fusion manner, while the XLM-R + CLIP architecture represents a dual-encoder late-fusion paradigm leveraging strong unimodal pretraining. This focused comparison enables clearer analysis of fusion strategy effectiveness without confounding factors introduced by multiple model variants.

5. Experimental Setup

5.1. Implementation Details

All models are implemented using PyTorch and the HuggingFace Transformers framework. Images are processed using the CLIP image processor, which performs resizing and normalization before being passed to the CLIP ViT-B/32 encoder. Meme text is tokenized using the corresponding XLM-RoBERTa tokenizers. Training is performed on a single GPU using mixed-precision (Automatic Mixed Precision, AMP) to improve computational efficiency and reduce memory usage.

5.2. Training Configuration

Subtask A: Hate Speech Detection. The XLM-RoBERTa-Base + CLIP late-fusion model is trained using the AdamW optimizer with a cosine learning-rate scheduler and linear warm-up. The learning rate is set to 2×10^{-5} with weight decay of 0.01.

To stabilize training, both text and image encoders are frozen during the first two epochs, allowing the classifier head to learn task-specific representations before full joint fine-tuning. Training employs automatic mixed precision and gradient clipping. Early stopping is applied based on validation Macro-F1. To address class imbalance, we perform validation-based threshold tuning and select

Component	Configuration
Text encoder (Subtask A)	XLM-R Base
Text encoder (Subtask B)	XLM-R Large
Image encoder	CLIP ViT-B/32
Optimizer	AdamW
Learning rate (head)	2×10^{-5}
Learning rate (encoders, B)	1×10^{-6}
Weight decay	0.01
Scheduler	Cosine (A), Linear (B) + warm-up
Batch size	16 (A), 8 (B)
Gradient accumulation	4 steps (Subtask B)
Epochs	10 (A), 20 (B)
Freeze epochs	2 (A), 5 (B)
Mixed precision	Automatic Mixed Precision (AMP)
Imbalance handling (A)	Validation threshold tuning
Imbalance handling (B)	Weighted sampler + Soft Focal Loss
Gradient clipping	1.0

Table 5: Training configuration of the proposed multimodal systems.

the probability threshold that maximizes Macro-F1 for final test prediction.

Subtask B: Sentiment Classification. For sentiment classification, we adopt a stronger XLM-RoBERTa-Large + CLIP fusion model due to the increased difficulty of three-class sentiment prediction. Training incorporates several stabilisation techniques designed for imbalanced and low-resource multimodal learning.

Balanced minibatches are generated using a WeightedRandomSampler, and Soft Focal Loss with class-dependent weighting is applied to improve minority-class recall without sacrificing precision. Encoders are frozen for the first five epochs and subsequently unfrozen for joint fine-tuning using discriminative learning rates for the classifier head and encoders. Gradient accumulation is applied to increase the effective batch size while maintaining GPU memory constraints. Early stopping is again based on validation Macro-F1.

Overall, the training strategy is designed to stabilize multimodal optimisation, mitigate class imbalance, and ensure fair comparison between early- and late-fusion architectures under limited shared-task data.

Due to the limited size of the shared-task dataset, we did not perform a full ablation study of individual training strategies. The stabilisation techniques described above follow widely adopted best practices for imbalanced multimodal classification and were applied consistently across experiments to ensure stable and fair comparison between fusion architectures.

6. Results and Analysis

6.1. Main Results

Table 6 summarizes the performance of the proposed systems on the shared task test set.

The results clearly show that the **XLM-RoBERTa + CLIP late-fusion model** outperforms the **ViLT + NepBERTa early-fusion baseline** on both subtasks. For Subtask A, the fusion model achieves a Macro-F1 of **0.6564**, improving over the ViLT baseline by more than 3.6 points. For Subtask B, the improvement is larger, with the fusion model achieving a Macro-F1 of **0.4859** compared to 0.4042 for the baseline.

6.2. Leaderboard Submission

We submitted our best-performing late-fusion systems to the official CHiPSAL 2026 evaluation server. Our system obtained the following official rankings:

- **Subtask A (Hate Speech Detection): Rank 7**
- **Subtask B (Sentiment Classification): Rank 8**

These rankings indicate that the proposed late-fusion approach provides a competitive and stable baseline under shared-task conditions.

6.3. Subtask Comparison

Hate speech detection consistently outperforms sentiment classification across all evaluated models. This difference reflects the intrinsic complexity of the tasks. Hateful memes often contain relatively explicit lexical or visual indicators (e.g., offensive wording or stereotypical imagery), enabling clearer decision boundaries. In contrast, sentiment in memes is frequently conveyed through humor, irony, exaggeration, or culturally grounded references, making polarity detection inherently more ambiguous. The performance gap between Subtask A and Subtask B therefore highlights the increased semantic complexity of multimodal sentiment understanding.

6.4. Early vs Late Fusion

The comparison between ViLT (early fusion) and XLM-R + CLIP (late fusion) reveals a consistent advantage for late-fusion modeling in this dataset setting.

The early-fusion ViLT architecture processes image and text jointly within a single vision–language transformer backbone, enabling rich cross-modal interaction. However, such joint transformers typically require substantial task-specific data to effectively learn cross-modal alignment. Under limited shared-task data, early-fusion models may struggle to fully exploit multimodal interactions.

To better support Nepali text, we replace the default tokenizer with the NepBERTa tokenizer and resize the ViLT embedding layer accordingly. The

Model	Subtask A				Subtask B			
	Acc	Pre	F1	Rec	Acc	Pre	F1	Rec
XLM-RoBERTa + CLIP	**0.7164**	0.6741	**0.6564**	0.6495	**0.4887**	0.4858	**0.4859**	0.5037
NepBERTa + ViLT	0.6493	0.6186	0.6203	0.6285	0.4286	0.4053	0.4042	0.406

Table 6: Performance metrics across Subtask A (Hate Speech Detection) and Subtask B (Sentiment Classification).

transformer encoder itself remains the original ViLT backbone.

In contrast, the late-fusion approach leverages two independently pretrained encoders:

- XLM-RoBERTa for multilingual text representation
- CLIP for robust visual feature extraction

By combining strong unimodal representations learned from large-scale pretraining, the late-fusion model benefits from stable features before multimodal classification. Our findings therefore suggest that late-fusion architectures are particularly effective under low-resource multimodal conditions, rather than implying a universal superiority over early fusion.

6.5. Impact of Multilingual Text Modeling

The strong performance of XLM-R based models highlights the importance of multilingual pretraining for Nepali meme understanding. Qualitative inspection reveals frequent English–Nepali code-mixing and script mixing between Devanagari Nepali and Roman-script English. Many memes include English lexical insertions (e.g., “Free WiFi”, “best friend”, page tags such as “SARCASM” or “TROLL”) embedded within Nepali sentences. These mixed-script patterns can be challenging for monolingual models and help explain why multilingual encoders capable of handling cross-script tokenization provide consistent advantages.

6.6. Error Analysis

Despite the improvements achieved by the late-fusion model, several recurring error patterns were observed:

- **Sarcasm and irony:** Memes explicitly marked with page-level cues such as “SARCASM” or “TROLL” are sometimes misclassified when the literal OCR text appears neutral or positive.
- **Implicit sentiment and cultural context:** Certain memes rely heavily on shared cultural knowledge or visual humor, making polarity difficult to infer from text alone.
- **Code-mixed and noisy tokens:** Frequent mixing of Devanagari Nepali with Roman-script English, along with URLs, hashtags, and page tags, introduces lexical variability that weakens token-level sentiment cues.

6.7. Discussion

Overall, the findings underscore the importance of multilingual pretraining and strong unimodal representations for multimodal meme understanding in low-resource settings. While hate speech detection benefits from more explicit lexical and visual cues, sentiment classification requires deeper multimodal reasoning and improved cross-modal alignment. These results suggest that future work should explore larger multilingual multimodal pretraining and more data-efficient early-fusion architectures for low-resource meme understanding.

7. Conclusion

In this paper, we presented our systems for the CHiPSAL 2026 shared task on multimodal hate and sentiment understanding in Nepali memes. We conducted a focused comparison between early-fusion (ViLT + NepBERTa) and late-fusion (XLM-RoBERTa + CLIP) multimodal architectures and demonstrated that the late-fusion dual-encoder model consistently achieves superior performance across both subtasks.

Our results highlight the importance of multilingual pretraining for handling real-world Nepali meme content, which frequently exhibits script-level code-mixing between Devanagari Nepali and Roman-script English. While hate speech detection achieved moderate performance, sentiment classification remains considerably more challenging due to sarcasm, humor, implicit emotional cues, and culturally grounded references.

Overall, our findings indicate that multilingual dual-encoder architectures provide a strong and stable baseline under limited shared-task data conditions. Future work will explore improved cross-modal attention mechanisms, dedicated code-mixed tokenization strategies, larger multilingual multimodal pretraining, and data augmentation techniques to further enhance performance in low-resource meme understanding.

Ethical Considerations

This work focuses on multimodal content moderation in Nepali memes, which may contain offensive, hateful, or sensitive material. The dataset used in this shared task was released by the organizers for research purposes, and we strictly followed the data usage guidelines provided with the shared task. No additional data collection or annotation involving human participants was performed by the authors.

We acknowledge that automated hate speech detection systems can introduce biases and may incorrectly label benign or culturally specific expressions as harmful. Such errors may disproportionately affect certain communities or linguistic groups. Therefore, the proposed models are intended solely for research and benchmarking purposes and should not be deployed in real-world moderation systems without thorough human oversight.

Additionally, the presence of script-level code-mixing (Devanagari Nepali with Roman-script English) introduces variability that may not be fully captured by current tokenization strategies. Future work should explore dedicated code-mixed tokenizers and multilingual multimodal pretraining for South Asian languages.

The goal of this work is to support safer online spaces and to advance research in low-resource languages, while recognizing the importance of fairness, transparency, and responsible deployment of automated moderation technologies.

Limitations

This work has several limitations. First, the dataset size is relatively small compared to large-scale multimodal datasets, which restricts the ability of deep models to fully generalize. Second, the models rely on pretrained multilingual encoders that may not fully capture cultural nuances, regional humor, or evolving internet slang in Nepali memes.

Third, sentiment classification results remain modest, highlighting the difficulty of modeling sarcasm, irony, and implicit emotional cues in memes. Many memes convey meaning through subtle visual symbolism or cultural context that current models cannot fully interpret.

Finally, the experiments focus on benchmark performance within a shared task setting. Additional work is required to evaluate robustness, fairness, and cross-domain generalization before real-world deployment.

References

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pages 5583–5594. PMLR.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and

sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246.

HasNat@CHiPSAL 2026: Multimodal Hate Speech Detection in Low-Resource Nepali Memes Using Aligned Vision–Language Models

Alvee Hasan Chowdhury, Md. Abul Hasnat, Adnan Faisal
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Bangladesh
{u2004011, u2004005, u2004002}@student.cuet.ac.bd

Abstract

Memes, while widely used for humor and everyday expression, have increasingly been exploited to spread hate and harmful stereotypes on social media. Detecting such content has been particularly challenging because meme meaning has been conveyed jointly through images and embedded text and the problem has become harder in low-resource languages like Nepali where annotated multimodal resources have remained limited. In the CHiPSAL 2026 Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes (Subtask-A), we have focused on identifying hate speech in Nepali-only memes by modeling both textual and visual cues. We have benchmarked three multimodal model families-ViT-B-32 (OpenCLIP), AltCLIP and BLIP2+mT5-under baseline, preprocessing, augmentation and combined settings. Among all systems, an AltCLIP-based late-fusion classifier has achieved the best validation performance with a macro F1-score of 0.66, indicating that aligned vision-language representations have been effective for meme-level understanding in Nepali. ViT-B-32 has provided a strong Baseline with a macro F1-score of 0.65, while BLIP2+mT5 has performed best when preprocessing and augmentation have been combined, achieving a macro F1-score of 0.62. Overall, our findings have highlighted the importance of multimodal modeling and meme-safe transformations for robust hate speech detection in low-resource meme content.

1 Introduction

Memes have been widely used on social media for humour and cultural commentary, typically combining an image with embedded text (Ramamoorthy et al., 2022). However, the same format has increasingly been exploited to spread hate and misinformation, making automated detection essential. Hate speech detection has remained challenging because it is often subtle, context-dependent and may require world knowledge (Kiela et al., 2021). While progress has been made for high-resource languages such as English, low-resource languages like Nepali have remained underexplored despite the growing presence of Nepali and code-mixed meme content (Thapa et al., 2025a).

The CHiPSAL 2026 Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes has focused on Nepali memes and has required models to jointly analyze both the image and its embedded Nepali text (Thapa et al., 2026a; Sarveswaran et al., 2026). In this work, we have evaluated three multimodal model families: ViT-B-32 (OpenCLIP), AltCLIP, and BLIP2+mT5. We compared these models under baseline, preprocessing, augmentation, and combined settings. AltCLIP (Baseline) has achieved the best validation performance with an F1-score of 0.66, highlighting the effectiveness of aligned vision-language representations for Nepali meme hate-speech detection. We have also examined preprocessing and augmentation variants to study how input cleaning and synthetic diversity have influenced robustness and generalization across backbones.

The core contributions of our research work have been as follows:

- We benchmarked multiple multimodal backbones (OpenCLIP, AltCLIP, BLIP2+mT5) under consistent experimental settings for Nepali meme hate speech detection.
- We analyzed the impact of preprocessing and augmentation variants on each backbone to study robustness and generalization.
- We identified AltCLIP (Baseline) as the strongest model in our experiments, achieving the highest validation macro F1-score of 0.66.

Detailed implementation information has been made available in the GitHub repository: https://github.com/alvee-hasan/Nepali_Meme_Hate_Speech_Notebooks

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 237–243
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

2 Related Work

Hate speech detection has been widely studied and recent work has emphasized that online harm is often multimodal, where text and images jointly convey abusive meaning. Benchmarks such as the Hateful Memes Challenge have driven progress on models that must reason over both modalities (Kiela et al., 2020, 2021). In parallel, shared tasks and datasets on meme sentiment and emotion understanding (e.g., Memotion) and multimodal meme sentiment resources (e.g., MemoSen) further highlighted the importance of combining visual and textual cues for robust prediction (Sharma et al., 2020; Ramamoorthy et al., 2022). Beyond hate, multimedia abuse benchmarks such as MAMI expanded the evaluation landscape to related harms in memes, reinforcing the need for multimodal modeling approaches (Fersini et al., 2022).

For low-resource and multilingual settings, cross-lingual transfer has been explored to reduce annotation costs and improve generalization when labeled data is limited (Bigoulaeva et al., 2021). In the Nepali context, prior work has released annotated hate speech data and analyzed hate phenomena in Nepali discourse, providing foundations for building moderation systems in this language (Thapa et al., 2023). Recently, multimodal prompt- based approaches have also been investigated for code-mixed and low-resource memes, suggesting that strong pretrained multimodal representations can be adapted effectively with careful task design (Thapa et al., 2025a,b). Building on these directions, the CHiPSAL 2026 shared task focuses on multimodal hate and sentiment understanding in Nepali-only memes, encouraging systems that jointly model meme images and Nepali text for hate speech detection (Subtask A) and sentiment analysis (Subtask B) (Thapa et al., 2026a; Sarveswaran et al., 2026). We also note that the annotation schema used in this shared-task data follows prior multimodal hate resources, including CrisisHateMM (Bhandari et al., 2023).

3 Data Description

The Nepali Meme Hate Speech Detection dataset (Thapa et al., 2026a,b; Bhandari et al., 2023) consists of paired meme images and Nepali text, organized into Train, Development and Test splits. It contains 1,068 training samples, 133 development samples and 134 test samples, with 1,335 corresponding images in total. The split-wise distribution is shown in Table 1.

Label	Train	Development	Test
Hate	720	98	–
Non-Hate	348	35	–
Total	**1068**	**133**	**134**

Table 1: Label-wise Distribution of Training, Development and Test Data

3.1 Data Augmentation

Data augmentation was explored only in the AltCLIP variants that explicitly included augmentation. In the baseline AltCLIP setting, no augmentation was applied. Images were loaded, converted to RGB and passed directly to the AltCLIP processor, while text was used in its original form after basic safe handling. This preserved the original meme layout, caption readability and image-text alignment. Among all AltCLIP variants, this baseline achieved the best overall performance, with 0.66 precision, 0.66 recall, 0.70 accuracy and 0.66 macro F1, suggesting that preserving the original meme structure was most beneficial.

In the augmentation-only AltCLIP setting, augmentation was applied only to images; no textual augmentation was used. The image pipeline consisted of RandomResizedCrop, RandomHorizontalFlip, ColorJitter and RandomRotation. Training data were also balanced by oversampling each class to 1000 samples per fold. Although this variant achieved high precision (0.85), recall dropped to 0.55 and macro F1 fell to 0.49. This indicates that stronger visual transforms likely distorted meme-specific cues, especially embedded text and fine caption layout.

In the preprocessing + augmentation AltCLIP setting, augmentation remained image-only but was made more conservative. The training pipeline used slight upscaling followed by random cropping, RandomHorizontalFlip and light ColorJitter, while random rotation and stronger distortions were avoided to better preserve text readability. This setting also used balanced oversampling, with 1200 samples per class per fold. Even so, it remained below the baseline, reaching 0.62 precision, 0.60 recall, 0.68 accuracy and 0.61 macro F1.

Overall, the AltCLIP results suggest that augmentation was not beneficial for this task. Because meme classification depends heavily on caption visibility, text placement and precise image-text correspondence, even moderate visual transformations may remove or weaken important task-relevant in-

formation.

4 Methodology

4.1 Problem Formulation

The task is to classify a meme as *Hate* or *Non-Hate* using multimodal Nepali meme data(Table 1). Given a meme instance $m = (t, i)$, where t is the embedded Nepali text and i is the associated image, we learn a mapping:

$$f(t, i) \rightarrow \{0, 1\},$$

where 0 denotes Non-Hate and 1 denotes Hate. The objective is to jointly model textual and visual cues so that implicit and explicit hate expressions can be detected more reliably than with a single modality.

4.2 Data Preprocessing

For text, missing values were replaced with empty strings and all entries were converted to string format. In the baseline AltCLIP setting, preprocessing was intentionally light so that the model could preserve the original meme wording, spelling variation, punctuation and code-mixed cues. The text was then tokenized using the AltCLIP processor with a maximum sequence length of 77, applying truncation and fixed-length padding. No transliteration was applied; Nepali text was retained in its original Devanagari form.

For the AltCLIP variants that used additional preprocessing (i.e., the *Preprocessing* and *Preprocessing + Augmentation* settings), a stronger text normalization pipeline was applied before tokenization. This pipeline included Unicode normalization, normalization of Devanagari digits to standard ASCII digits, lowercasing of Latin-script characters, regex-based removal of source-page and watermark phrases, cleanup of URLs and user handles, whitespace normalization, repetition cleanup for unusually noisy OCR strings and text-length capping. These steps were designed primarily to reduce OCR noise, suppress source leakage and standardize noisy user-generated text while preserving the main semantic content of the meme.

For images, each meme was converted to RGB and resized/normalized using the AltCLIP image processor at a resolution of 224×224. In the baseline AltCLIP model, this processor-driven pipeline was used directly. In the preprocessing-based AltCLIP variants, an additional edge-cropping step was applied before resizing in order to suppress border watermarks and page-name overlays that frequently appeared near the image margins. We also verified image availability prior to training to prevent broken samples; if image loading failed, a white fallback image was used to ensure robustness. The *Preprocessing + Augmentation* AltCLIP variant inherited the same preprocessing pipeline before any training-time augmentation was applied.

To improve reproducibility and stability, we set fixed random seeds and employed a stratified split strategy during cross-validation so that class proportions remained consistent across folds. For completeness, the other explored backbones followed broadly similar goals, namely OCR-noise reduction and watermark suppression. ViT-B/32 used a comparable CLIP-style preprocessing pipeline, whereas the BLIP-2 + mT5 variants additionally used stronger text cleaning such as repeated n-gram filtering and a separate mT5-based text encoding branch. But overall, AltCLIP (Baseline) was the best-performing model.

4.3 Multimodal Model

AltCLIP was used as the multimodal backbone. The model encodes text and images separately, producing aligned text and image embeddings. These embeddings are then concatenated and passed to a trainable MLP classifier (late fusion).

Let $\mathbf{e}_t \in \mathbb{R}^d$ and $\mathbf{e}_i \in \mathbb{R}^d$ denote the text and image embeddings, respectively. The fused representation is computed as:

$$\mathbf{z} = [\mathbf{e}_i; \mathbf{e}_t] \in \mathbb{R}^{2d},$$

followed by:

$$\hat{y} = \sigma(\text{MLP}(\mathbf{z})).$$

The AltCLIP backbone was kept frozen and only the fusion classifier was trained. This design reduced memory cost and supported stable training while retaining strong pretrained cross-modal alignment.

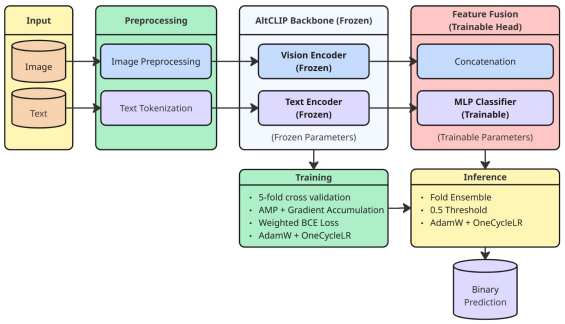


Figure 1: Overview of the multimodal framework for Nepali meme hate speech detection.

4.4 Training Setup

The model was trained on the combined Train and Development sets using 5-fold stratified cross-validation to preserve class distribution across folds. Optimization was performed using AdamW (learning rate 1×10^{-3}, weight decay 0.01) for 6 epochs per fold, with a batch size of 8 and gradient accumulation over 2 steps to maintain memory efficiency. To address class imbalance, we applied weighted binary cross-entropy with fold-specific class weights computed from the training labels. For stable and efficient learning, we used mixed precision (AMP), gradient clipping (max norm 1.0) and a OneCycle learning-rate schedule. During inference, we averaged the predicted probabilities from the best checkpoint of each fold (fold ensemble) and applied a 0.5 threshold for final label prediction.

4.5 Evaluation Metrics

The models were evaluated using macro-F1 score, precision and recall to ensure balanced performance and accurate identification of hate speech in Nepali-only memes.

5 Results and Analysis

This task evaluates multimodal models using meme text and images for Nepali hate-speech detection. While several settings achieved strong validation scores, the results indicate limited generalization, motivating more robust fusion and meme-aware augmentation.

5.1 Task: Hate Speech Detection in Nepali-only Memes

Model	Setting	P	R	F1
ViT-B-32	Preprocessing	0.62	0.60	0.61
	Augmentation	0.61	0.61	0.61
	Preprocessing + Augmentation	0.64	0.63	0.63
	Baseline	**0.65**	**0.64**	**0.65**
AltCLIP	Preprocessing	0.63	0.62	0.62
	Augmentation	**0.85**	0.55	0.49
	Preprocessing + Augmentation	0.62	0.60	0.61
	Baseline	0.66	**0.66**	**0.66**
BLIP2+mT5	Preprocessing	0.60	0.57	0.57
	Augmentation	0.59	0.58	0.58
	Preprocessing + Augmentation	**0.63**	**0.62**	**0.62**
	Baseline	0.61	0.61	0.61

Table 2: Performance comparison across model families and settings. Best results per model family are highlighted in bold.

Table 2 reports the performance of three multimodal backbones: ViT-B-32, AltCLIP, and BLIP2+mT5. The models are evaluated under four data settings: baseline, preprocessing, augmentation, and preprocessing with augmentation. For the ViT-B-32 family, the baseline configuration yields the strongest results with P=0.65, R=0.64 and F1=0.65, indicating that the default training setup already provides a robust CLIP-based baseline for Nepali meme hate-speech detection. In the AltCLIP experiments, the baseline achieves the best overall performance within the family (P=0.66, R=0.66, F1=0.66), demonstrating improved alignment of visual and textual representations compared to ViT-B-32. Although the augmentation-only AltCLIP setting achieved the highest precision (0.85), its substantially lower recall (0.55) reduced the overall F1-score, indicating that many positive memes were still missed. This was likely caused by the combination of residual class imbalance and an aggressive augmentation setup, including random resized cropping, horizontal flipping, color jitter and rotation, together with a very low learning rate, which may have weakened fine-grained meme text cues and reduced sensitivity. These transforms likely made the model more conservative by weakening subtle meme-specific cues, especially embedded text and local caption structure. As a result, the model tended to detect only the clearest hateful cases, which increased precision but reduced recall. A milder setup with gentler cropping, little or no flipping or rotation, lighter color jitter and a slightly higher learning rate would likely improve recall. For BLIP2+mT5, the preprocessing + augmentation configuration performs best (P=0.63, R=0.62, F1=0.62), showing that combining input cleaning with augmentation is beneficial for this architecture. Overall, AltCLIP provides the most balanced performance among the compared families, with the baseline achieving the best results, while augmentation-only causes the largest drop. Although preprocessing + augmentation recovers some performance, it remains below the unaugmented baseline.

5.2 Parameter Setting

Table 3 has summarized the main hyperparameters used in our experiments. For ViT-B-32, we report the learning rates for the two training stages (Stage-1 linear probing and Stage-2 light fine-tuning) in separate rows to keep the table compact. AltCLIP has been trained in a head-only setting with

the backbone frozen for memory efficiency. For BLIP2+mT5, we used a two-step pipeline and reported separate batch sizes for feature extraction and classifier training.

Model	Learning Rate	Optimizer	Batch Size
ViT-B-32 – Stage 1	3×10^{-3}	AdamW	32
ViT-B-32 – Stage 2	1×10^{-5}	AdamW	32
AltCLIP (head-only)	1×10^{-3}	AdamW	8
BLIP2+mT5 (extract)	3×10^{-4}	AdamW	4
BLIP2+mT5 (classifier)	3×10^{-4}	AdamW	64

Table 3: Key hyperparameters for the three model families. ViT-B-32 uses a two-stage training strategy with separate learning rates.

6 Conclusion

In this work, we addressed CHiPSAL 2026 Subtask-A on hate speech detection in Nepali-only memes by evaluating three multimodal model families: ViT-B-32 (OpenCLIP), AltCLIP, and BLIP2+mT5. We compared these models under baseline, preprocessing, augmentation, and combined settings. Among all systems, the AltCLIP baseline achieved the best validation performance, confirming the effectiveness of strongly aligned vision-language embeddings for low-resource meme moderation. We further observed that data-centric strategies were highly model-dependent: preprocessing and augmentation sometimes reduced performance when they removed dataset-specific cues (e.g., watermarks) or distorted embedded text, leading to recall drops. In particular, AltCLIP performed best without image augmentation, while aggressive visual transforms hurt meme classification by weakening embedded text and local caption layout; milder combined augmentation reduced this damage but still underperformed the baseline. Overall, our results highlight the importance of multimodal representations and meme-safe transformations for reliable Nepali hate-speech detection. Partial unfreezing of the top AltCLIP layers may improve macro F1 by enabling limited task-specific adaptation, although the small dataset size makes overfitting a significant risk. Future work will therefore examine threshold calibration, controlled ablations, and carefully limited backbone fine-tuning to improve robustness and generalization for practical content moderation support.

Error Analysis

Although AltCLIP (Baseline) performs best overall, the augmentation-based results show that simple count balancing did not fully solve the data skew problem. Table 4 summarizes how class balancing differed across the augmentation-based settings and helps explain why oversampling did not consistently improve generalization. ViT-B/32 augmentation variants remained imbalanced because no resampling was applied, while AltCLIP and BLIP-2 balanced only the training fold or train set through oversampling, mainly inflating the minority Non-Hate class without increasing unique example diversity. As a result, the experiments remained effectively imbalanced at evaluation time, since validation still followed the original class prior. This is especially problematic for meme classification, where embedded text, caption placement, and image-text alignment carry much of the signal, so aggressive transforms can easily damage critical cues. Even beyond this, AltCLIP still makes errors on memes containing implicit or context-dependent hate, such as sarcasm, idioms, or cultural cues that go beyond literal wording. It also fails when image-embedded text is small, stylized, or partially hidden, leading to missed cues and false negatives-an issue that worsens with stronger transforms like cropping or blur that reduce readability. Moreover, score differences across preprocessing and augmentation settings are partly affected by non-identical training conditions, including batch size, learning rate, and sampling strategy, while class balancing can also alter calibration; therefore, a fixed 0.5 threshold may misclassify borderline cases and hurt macro-F1.

Limitations

Despite strong performance from the Baseline AltCLIP system, limitations remain. The dataset is imbalanced and contains source-specific artifacts (e.g., watermarks), which may encourage shortcut learning; removing them can also discard label-correlated cues. The backbone was kept mostly frozen, limiting adaptation to distribution shifts introduced by preprocessing or augmentation. Cultural context and implicit intent in memes are also difficult to capture with simple embedding fusion and thresholds were not tuned separately for each setting, so calibration shifts may affect macro-F1. For BLIP2+mT5, hardware limitations on the Tesla P100 imposed a major constraint. End-to-end BLIP-2 fine-tuning was infeasible because the

Models	Training Scope	Label 0 (Non-Hate)	Label 1 (Hate)	Reason of augmentation failure
ViT-B/32 Augmentation	combined train+val for 5-fold CV	383 → 383	818 → 818	No resampling, still imbalanced
ViT-B/32 Preprocessing + Augmentation	combined train+val for 5-fold CV	383 → 383	818 → 818	No resampling, still imbalanced
AltCLIP Augmentation	per CV training fold	306–307 → 1000	654–655 → 1000	Fold-only oversampling balance
AltCLIP Preprocessing + Augmentation	per CV training fold	306–307 → 1200	654–655 → 1200	Fold-only oversampling balance
BLIP-2 Augmentation	train only	348 → 1200	720 → 1200	Train-only oversampling balance
BLIP-2 Preprocessing + Augmentation	train only	348 → 1200	720 → 1200	Train-only oversampling balance

Table 4: Class-balancing behavior in augmentation-based settings.

full model exceeded memory limits, while 8-bit quantization could not be used due to bitsandbytes incompatibility with the P100 architecture. As a result, BLIP-2 was used only as a frozen feature extractor with a lightweight classifier, which reduced memory usage but likely capped task-specific adaptation and overall performance.

Ethical Statement

This work followed established ethical guidelines for handling sensitive hate-speech content. It aimed to improve automated hate-speech detection while respecting user privacy and fundamental rights. We acknowledge the possibility of dataset bias and label subjectivity and we have reported the limitations of our study transparently. The proposed model is intended to support human moderation rather than replace human judgment.

Acknowledgement

We have sincerely thanked the organizers of the CHiPSAL 2026 Shared Task for providing the benchmark dataset and evaluation framework. We have acknowledged (Thapa et al., 2025a) for releasing the NeMeme dataset, which forms the foundation of this study. We have expressed our sincere gratitude to the open-source community for developing and openly sharing the tools, libraries and pretrained models that have made this work possible.

References

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Irina Bigoulaeva, Viktor Hangya, and Alexander Fraser. 2021. Cross-lingual transfer learning for hate speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 15–25, Kyiv. Association for Computational Linguistics.

Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. 2022. SemEval-2022 task 5: Multimedia automatic misogyny identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 533–549, Seattle, United States. Association for Computational Linguistics.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Casey A. Fitzpatrick, Peter Bull, Greg Lipstein, Tony Nelli, Ron Zhu, Niklas Muennighoff, Riza Velioglu, Jewgeni Rose, Phillip Lippe, Nithin Holla, Shantanu Chandra, Santhosh Rajamanickam, Georgios Antoniou, Ekaterina Shutova, Helen Yannakoudakis, Vlad Sandulescu, Umut Ozertem, Patrick Pantel, Lucia Specia, and Devi Parikh. 2021. The hateful memes challenge: Competition report. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 344–360. PMLR.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020), December 6–12, 2020, virtual*.

Sathyanarayanan Ramamoorthy, Nethra Gunti, Shreyash Mishra, Suryavardan S, Aishwarya N. Reganti, Parth Patwa, Amitava Das, Tanmoy Chakraborty, Amit P. Sheth, Asif Ekbal, and Chaitanya Ahuja. 2022. Memotion 2: Dataset on sentiment and emotion analysis of memes (short paper). In *Proceedings of the Workshop on Multi-Modal Fake News and Hate-Speech Detection (DE-FACTIFY 2022), co-located with AAAI 2022*, volume 3199 of *CEUR Workshop Proceedings*.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Kr-

ishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 task 8: Memotion analysis- the visuo-lingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online). International Committee for Computational Linguistics.

Surendrabikram Thapa, Kritesh Rauniyar, Shuvam Shiwakoti, Sweta Poudel, Usman Naseem, and Mehwish Nasim. 2023. Nehate: Large-scale annotated data shedding light on hate speech in nepali local election discourse. In *ECAI 2023 - 26th European Conference on Artificial Intelligence*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 2346–2353. IOS Press.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026a. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Surendrabikram Thapa et al. 2026b. therealthapa/chipsal26-memes: Shared task on multimodal hate and sentiment understanding in low-resource memes (chipsal 2026). GitHub repository. Accessed: 2026-02-27.

TeamHerald@CHIPSAL 2026: Hate Speech Detection and Sentiment Analysis of Nepali Memes using Transformer-based Architectures and Ensemble Learning

Ashish Acharya*, **Anish Khatiwada***, **Rohit Khadka**, **Pragya Aryal**
Herald College Kathmandu, Kathmandu, Nepal
{ashish.acharya048, anishkhatioda, rohitkhadka153, pragyaaryal18}@gmail.com
*These authors contributed equally to this work.

Abstract

The analysis of internet memes in the Nepali language is complicated by frequent code-mixing and a lack of established baseline resources. While memes inherently combine visual and textual elements, this study focuses on a text-centric approach by extracting embedded text using an OCR layer and modeling it with Transformer-based architectures. We evaluate six distinct models and investigate the comparative effectiveness of Hard and Soft Voting ensemble strategies across two tasks: binary hate speech detection and three-class sentiment analysis. Experimental results show that a standalone decoder-only model achieved the highest performance for binary classification, whereas the Soft Voting ensemble performed best for the multi-class sentiment task, yielding a 15.8% relative improvement in Macro F1-score over the strongest standalone baseline. These findings suggest that ensemble strategies behave differently across binary and multi-class tasks, highlighting the importance of selecting aggregation methods suited to the classification objective.

Keywords: Nepali Language Processing, Meme Analysis, Ensemble Methods, Code-Mixed Text

1. Introduction

Internet memes have evolved from simple units of humor into sophisticated instruments for socio-political discourse, cultural expression, and, increasingly, the dissemination of harmful content. Unlike traditional text-only social media posts, memes are inherently multimodal, with meaning emerging from the interaction between visual imagery and embedded text (Kiela et al., 2020). This combination often creates compositional effects, where the sentiment of the text can be inverted or intensified by the accompanying image, rendering unimodal detection systems largely ineffective (Pramanick et al., 2022).

For low-resource languages such as Nepali, these challenges are compounded. The linguistic landscape features frequent code-mixing, including Devanagari and Romanized Nepali alongside English, which complicates standard tokenization and semantic analysis (Rauniyar et al., 2023). In addition, there is a scarcity of high-quality, annotated multimodal datasets that capture the cultural and political nuances of Nepali digital content. While significant progress has been made in English-centric meme analysis, such models often fail to generalize to South Asian contexts, where hate speech and sentiment are closely tied to local events, ethnic identities, and region-specific political satire.

To address these limitations, this paper evaluates a robust text-centric pipeline for Nepali meme analysis. An Optical Character Recognition (OCR) layer extracts textual content from raw images, simulating a real-world scenario in which ground-truth text is unavailable. We then benchmark six Transformer-based architectures, including models pre-trained on Nepali and cross-lingual corpora, to identify the most effective base learners for this domain.

The main contribution of this work is a systematic evaluation of ensemble strategies within a multimodal framework. By applying both Hard (majority-based) and Soft (probability-based) voting across binary and multi-class tasks, we examine how ensemble logic interacts with task complexity. Our analysis shows that Hard Voting stabilizes binary hate speech detection by mitigating model-specific noise, while Soft Voting is more effective for multi-class sentiment classification. This indicates that preserving the probabilistic confidence of underlying models is beneficial for resolving subtle linguistic ambiguities in multi-class social media discourse.

2. Related Work

Significant advances have been made in hate speech detection and sentiment analysis, particularly for high-resource languages such as English. However, low-resource languages, including Nepali, remain underrepresented in both dataset availability and model development (Parihar et al., 2021). The challenge is further amplified in the context of multimodal content such as memes, where meaning emerges from the interaction between textual and visual modalities.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 244–249
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

2.1. Hate Speech and Sentiment Analysis in Low-Resource Languages

Hate speech and sentiment analysis in low-resource languages has gained increasing attention due to the rapid expansion of South Asian social media content. However, compared to English, languages such as Nepali, Hindi, Bangla, and Hinglish present significant challenges due to limited annotated datasets, linguistic diversity, and frequent code-mixing.

Early sentiment analysis work in Nepali relied on traditional machine learning models such as Support Vector Machines (SVM) and Naive Bayes with hand-crafted features (Sitaula et al., 2021). These approaches demonstrated limited generalization capability. The emergence of deep learning enabled the use of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, which improved performance through contextual embeddings. (Joshi et al., 2019)

For Bangla and Hindi, several studies have investigated abusive and hate speech detection in social media, emphasizing issues such as informal spelling, transliteration, and cultural nuance (Das and Mukherjee, 2021). Transformer-based models, particularly multilingual and monolingual pretrained architectures, have consistently demonstrated improved performance in these settings (Rathore and Chaurasia, 2025; Das et al., 2022).

However, recent findings suggest that domain-relevant non-hateful pretraining data may yield performance improvements comparable to explicitly hateful pretraining, challenging assumptions about task-specific pretraining strategies (Gokhale et al., 2022). Despite progress, data scarcity and domain adaptation remain core challenges in low-resource setups.

2.2. Multimodal Hate and Meme Analysis

With memes becoming a dominant medium for political expression and social commentary, hate speech detection has expanded beyond text-only analysis to multimodal settings. Memes often encode meaning through the interaction of visual and textual components, making unimodal models insufficient (Ji et al., 2023).

Previous multimodal hate detection research shows that models integrating both visual and textual features outperform text-only approaches (Shah et al., 2024). In many cases, hateful or sarcastic intent is not explicit in either modality alone but emerges from their combination (Yadav et al., 2024). Election-related memes, particularly in South Asian contexts, have been shown to propagate hate and misinformation, requiring culturally informed modeling approaches (Jafri et al., 2024).

Prompt-based and contrastive learning methods have recently gained attention for modeling implicit cross-modal relationships without requiring extensive labeled datasets (Ranasinghe et al., 2021). CLIP-based architectures and adapter-based fine-tuning strategies further enhance multimodal meme classification, though detecting creative, humorous, or sarcastic hateful memes remains an open challenge, especially in multilingual and low-resource environments (Shah et al., 2024; Ji et al., 2023).

2.3. Datasets and Annotation Challenges

A major bottleneck in low-resource hate speech and sentiment analysis is the scarcity of high-quality annotated datasets. Creating datasets for languages such as Nepali, Hindi, and Bangla requires linguistic expertise and cultural understanding, particularly when addressing political or election-related discourse (Jafri et al., 2024).

Multimodal meme datasets introduce additional complexity, as annotators must interpret visual context, textual meaning, humor, sarcasm, and sociopolitical nuance simultaneously (Ji et al., 2023; Shah et al., 2024). Many existing South Asian hate speech datasets remain text-only and relatively small in scale (Das and Mukherjee, 2021; Gokhale et al., 2022).

Code-mixed datasets further complicate annotation due to ambiguous language boundaries, inconsistent spelling, and annotator disagreement (Rathore and Chaurasia, 2025). Studies frequently report moderate inter-annotator agreement for subjective labels such as hate and sarcasm (Ranasinghe et al., 2021). These challenges highlight the need for clearer annotation protocols and scalable dataset creation strategies in low-resource multimodal research.

2.4. Modeling Approaches

Modeling strategies for hate speech and sentiment classification have evolved from traditional bag-of-words and TF-IDF approaches to deep neural architectures (Sitaula et al., 2021; Das and Mukherjee, 2021). While traditional models are computationally efficient, they struggle to capture semantic and contextual nuance in informal social media text.

Recurrent neural networks (RNNs), LSTMs, and GRUs improved contextual modeling capabilities, particularly in Nepali and Hindi sentiment analysis (Joshi et al., 2019). Transformer-based architectures further advanced the field through large-scale pretraining and cross-lingual transfer learning (Rathore and Chaurasia, 2025; Das et al., 2022).

In multimodal scenarios, vision-language transformer models such as CLIP have shown strong performance by learning joint image-text representations (Shah et al., 2024). Prompt-based learning

and lightweight adaptation methods offer promising directions for low-resource meme classification tasks (Ranasinghe et al., 2021). Nevertheless, interpretability and cultural sensitivity of large transformer models in low-resource contexts remain underexplored.

Despite substantial progress, several gaps persist. First, limited work simultaneously addresses low-resource languages, code-mixing, and multimodal meme analysis within a unified framework. Second, Nepali multimodal meme datasets remain scarce, restricting comprehensive evaluation. Third, although transformer and CLIP-based models demonstrate strong empirical performance, their interpretability and socio-cultural alignment in South Asian contexts require further investigation. Finally, scalable annotation strategies and weakly supervised or prompt-based methods are necessary to reduce reliance on large labeled datasets.

Addressing these gaps is critical for developing robust hate speech and sentiment classification systems for multilingual, low-resource, and multimodal social media environments.

3. Datasets and Tasks Description

The shared task consists of two different subtasks: Subtask A aims to detect hate speech in Nepali meme images, Subtask B is related to classifying the sentiment from Nepali meme images. For all of the mentioned subtasks, datasets were provided by the organizers, which were initially created and curated by different papers (Thapa et al., 2026; Sarveswaran et al., 2026).

3.1. Sub-Task A

It involves binary classification to distinguish between images classified as HATE (labeled as 1) and NO-HATE (labeled as 0) (Thapa et al., 2025a,b; Bhandari et al., 2023). For the training purposes of this sub-task, a total of 1068 datasets were provided, out of which 720 were labeled as Hate and 348 were labeled as No Hate. The other 133 datasets were provided for validation, and 134 for testing purposes.

3.2. Sub-Task B

It involves multi-class classification to distinguish between images classified as Negative (labeled as 0), Neutral (labeled as 1), and Positive (labeled as 2) (Thapa et al., 2025a,b; Bhandari et al., 2023). For the training purpose of this sub-task, a total of 1061 datasets were provided, out of which 473 were labeled as Negative, 341 were labeled as Neutral and 247 were labeled as Positive. The other 133 data sets were for validation, and 133 for testing purposes.

4. Methodology

Our approach was designed to navigate the inherent multimodality of memes, where visual and textual cues are often inextricably linked. We developed a pipeline that first extracts embedded text via EasyOCR and subsequently processes these strings through a Transformer-based ensemble. While ground-truth text labels were available during the training phase, we intentionally restricted the testing phase to raw images. This design choice was made to simulate a real-world deployment scenario where OCR noise and extraction failures are inevitable (Rauniyar et al., 2023).

4.1. Dataset Balancing

A major challenge in our data pipeline was class imbalance, which can bias models toward majority labels and negatively impact Macro F1-score. To address this, we applied random oversampling to the minority classes to obtain a more balanced training distribution. Specifically, samples were resampled with replacement to reach $n = 1000$ instances per class for Subtask B and $n = 1500$ for Subtask A.

These target sizes were selected based on preliminary experiments, where increasing the representation of minority classes led to improved validation performance, particularly for Subtask A. This approach helps ensure that all classes contribute more evenly to the loss function during training, reducing bias toward majority classes.

4.2. Model Selection

To capture code-mixed and low-resource linguistic features, we fine-tuned six Transformer-based architectures from Hugging Face, each selected for specific advantages:

- **NepaliBERT**: BERT-base model specialized for native Nepali semantics, providing a robust baseline for formal and semi-formal Devanagari text.
- **DistilBERT-Base-Nepali**: Knowledge distilled, computationally efficient variant for Nepali text, included to evaluate performance trade-offs in resource-constrained environments.
- **XLM-RoBERTa-Base**: Multilingual model capable of handling complex code-mixed content by leveraging shared representations across 100 languages.
- **RoBERTa-Hindi**: Used for cross-lingual transfer learning; shared Devanagari script and lexical overlap between Hindi and Nepali may en-

hance feature representation for low-resource tasks.

- **distilgpt2-nepali**: Decoder-only GPT-2 model with continual pre-training on 13 million Nepali sequences. This model was evaluated only as a standalone baseline and was not included in the ensemble.

4.3. Experimental Setup

All models were fine-tuned using the parameters detailed in Table 1. We implemented **Early Stopping** with a patience of 3 epochs and a threshold of 0.01 to monitor validation loss.

Table 1: Fine-Tuning Hyperparameters

Hyperparameter	Value
Epochs	10
Batch Size	16
Learning Rate	2×10^{-5}
Weight Decay	0.01
Early Stopping Patience	3
Logging Steps	50

4.4. Evaluation Framework: Individual and Ensemble Testing

Following the evaluation of the six standalone models, we implemented an ensemble layer to aggregate their predictions. This was conducted using two distinct voting strategies applied across both subtasks to identify the most robust method for final classification.

4.4.1. Hard Voting

In the Hard Voting setup, the final prediction $\hat{y}$ is derived from a majority consensus of the five base learners. In this configuration, each model contributes a single discrete label, and the most frequent class is selected as the final output. This method treats all models as equal contributors regardless of their individual confidence scores.

4.4.2. Soft Voting

We concurrently tested a Soft Voting strategy, which aggregates the predicted class probabilities (softmax outputs) rather than discrete labels. The final predicted class $\hat{y}$ is calculated as:

$$\hat{y} = \arg\max_i \sum_{j=1}^{5} w_j p_{ij}$$

where p_{ij} represents the probability assigned to class i by model j. This allows the ensemble to account for the relative confidence of each base learner across the 2-class (Subtask A) and 3-class (Subtask B) problems.

5. Results and Analysis

The quantitative performance across both subtasks, summarized in Table 2, highlights architectural differences in low-resource multimodal NLP settings. In accordance with the official shared task evaluation criteria, Macro F1-score is reported as the primary metric.

In Subtask A (Hate Speech), the standalone Sakonii/distilgpt2-nepali model achieved the highest performance ($F1 = 0.6550$). This result may indicate that domain-specific continual pre-training contributed to improved binary discrimination between offensive and non-offensive content. Among the encoder-based models, RoBERTa-Hindi and NepaliBERT achieved identical evaluation metrics across all reported measures; however, their predictions differ at the instance level, suggesting similar class-wise error distributions. Although the ensemble approaches (Hard and Soft Voting) remained competitive, they did not surpass the GPT-2 baseline. This suggests that for binary classification tasks, a well-adapted single model can perform comparably to or better than aggregation-based methods.

In contrast, Subtask B (Sentiment Classification) revealed greater benefits from model aggregation under a three-class setting. The Soft Voting ensemble achieved the highest Macro F1-score (0.5518), representing a 15.8% relative improvement over the strongest standalone baseline (RoBERTa-Hindi). This pattern suggests that probabilistic aggregation may better capture uncertainty in multi-class sentiment classification compared to majority-based voting. Overall, these findings indicate that in low-resource settings, performance gains may depend not only on model selection but also on the aggregation strategy employed.

6. Limitations

While the proposed approach demonstrates performance gains for Nepali meme classification, several limitations should be considered when interpreting these findings.

First, the evaluation relies on a text-centric pipeline dependent on the EasyOCR extraction layer. Because the classification models operate solely on extracted text without incorporating visual features, overall performance is sensitive to OCR quality. A qualitative inspection of OCR outputs revealed frequent errors in stylized fonts, low-resolution text, and visually complex layouts, often resulting in incomplete or distorted tokens. These

Table 2: Experimental Results: Comparative Performance of Standalone Base Learners and Ensemble Strategies across Subtasks A and B.

Task	Model/Strategy	F1 Macro	Accuracy	Precision	Recall
Subtask A (Hate Speech)	Twitter-XLM-RoBERTa-Base	0.5830	0.6343	0.5836	0.5826
	DistilBERT-Base-Nepali	0.5318	0.5970	0.5332	0.5316
	RoBERTa-Hindi	0.5878	0.6343	0.5874	0.5884
	NepaliBERT	0.5878	0.6343	0.5874	0.5884
	XLM-RoBERTa-Base	0.5387	0.6343	0.5545	0.5419
	Sakonii/distilgpt2-nepali	**0.6550**	**0.7090**	**0.6652**	**0.6497**
	Hard Voting (Ensemble)	0.6422	0.6940	0.6482	0.6386
	Soft Voting (Ensemble)	0.6310	0.6866	0.6382	0.6273
Subtask B (Sentiment)	Twitter-XLM-RoBERTa-Base	0.4110	0.4286	0.4456	0.4333
	DistilBERT-Base-Nepali	0.4341	0.4586	0.4341	0.4352
	RoBERTa-Hindi	0.4764	**0.5789**	**0.6481**	0.4918
	NepaliBERT	0.3945	0.3985	0.3955	0.4083
	XLM-RoBERTa-Base	0.4484	0.4511	0.4484	0.4623
	Sakonii/distilgpt2-nepali	0.4108	0.4436	0.4222	0.4077
	Hard Voting (Ensemble)	0.4622	0.4887	0.5082	0.4635
	Soft Voting (Ensemble)	**0.5518**	0.5639	0.5659	**0.5452**

issues may lead to error propagation, where downstream classification is performed on noisy or partially incorrect inputs.

Second, the linguistic complexity of the dataset remains a persistent challenge. The prevalence of code-mixing (Nepali, English, and Romanized Nepali), along with colloquial expressions, may not be fully captured by existing pre-training corpora. This limitation is particularly evident in the three-class sentiment task, where subtle contextual and semantic distinctions are required to differentiate between neutral and polarized classes.

Finally, the use of random oversampling to balance class distributions creates a controlled training setup that does not reflect the naturally skewed distribution of hate speech in real-world social media. As a result, the reported performance should be interpreted within the context of balanced experimental conditions rather than as a direct estimate of real-world deployment performance.

7. Conclusion

We evaluated several Transformer architectures and ensemble strategies for hate speech and sentiment analysis in the low-resource Nepali language. The experimental results suggest differing strengths between standalone and ensemble architectures across tasks. While a standalone decoder-only model (Sakonii/distilgpt2-nepali) achieved the highest performance in binary hate speech detection, the Soft Voting ensemble achieved the strongest performance in the three-class sentiment task. Specifically, the 15.8% relative improvement in F1-Macro for Subtask B suggests that probabilistic aggregation may be beneficial for multi-class sentiment classification.

These findings indicate that performance in this task varies with class cardinality and aggregation strategy. While memes are inherently multimodal, this work adopts a text-centric approach based on OCR-extracted content, which may limit performance in cases where meaning is conveyed primarily through visual context. Future work will explore the integration of visual-semantic features, including models such as CLIP or Vision Transformers (ViT), to better capture image-text interactions. Additionally, the application of parameter-efficient fine-tuning techniques remains a promising direction to investigate the balance between the computational requirements of ensemble-based inference and classification accuracy.

8. Bibliographical References

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Mithun Das and Animesh Mukherjee. 2021. Banglaabusememe: A dataset for bengali abusive meme classification. In *Conference on Empirical Methods in Natural Language Processing*.

Mithun Das, Punyajoy Saha, Binny Mathew, and Animesh Mukherjee. 2022. HateCheckHIn:

Evaluating Hindi hate speech detection models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5378–5387, Marseille, France. European Language Resources Association.

Omkar Gokhale, Aditya Kane, Shantanu Patankar, Tanmay Chavan, and Raviraj Joshi. 2022. Spread love not hate: Undermining the importance of hateful pre-training for hate speech detection.

Farhan Jafri, Kritesh Rauniyar, Surendrabikram Thapa, Mohammad Siddiqui, Matloob Khushi, and Usman Naseem. 2024. Chunav: Analyzing hindi hate speech and targeted groups in indian election discourse. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Junhui Ji, Wei Ren, and Usman Naseem. 2023. Identifying creative harmful memes via prompt based approach. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 3868–3872, New York, NY, USA. Association for Computing Machinery.

Aditya Joshi, Ashutosh Upadhyay, and Shishir Kumar. 2019. Deep learning for Nepali sentiment analysis. *arXiv preprint arXiv:1911.00000*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2611–2624.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Shraman Pramanick, Shivam Sharma, Nikolaos Dimaridis, Akhtar Md Shad, Tanmoy Chakraborty, Preslav Nakov, and Dinesh Manocha. 2022. Momenta: A multimodal framework for detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4439–4455.

T. Ranasinghe, E. Zampieri, and M. Mihaela. 2021. Multilingual and multimodal approaches for hate speech detection. In *SemEval Workshop*.

B. S. Rathore and S. Chaurasia. 2025. Fine tuning large language models for hate speech detection in hinglish and code mixed custom dataset through a socially responsible approach for safer digital platforms. *Discover Sustainability*, 6:1409.

Krisha Rauniyar, Gaurav Paudel, Biraj Khanal, and Bal Krishna Bal. 2023. Detection of hate speech, targets, humor, and stance in Nepali memes. In *Proceedings of the 7th Workshop on Online Abuse and Harms (WOAH)*, pages 154–165. Association for Computational Linguistics.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Siddhant Shah, Shuvam Shiwakoti, Maheep Chaudhary, and Haohan Wang. 2024. Memeclip: Leveraging clip representations for multimodal meme classification.

C. Sitaula, A. Basnet, A. Mainali, and {T. B.} Shahi. 2021. Deep learning-based methods for sentiment analysis on nepali text. *Computational Intelligence and Neuroscience*.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Anjali Yadav, Tanya Garg, Matej Klemen, Matej Ulcar, Basant Agarwal, and Marko Robnik Sikonja. 2024. Code-mixed Sentiment and Hate-speech Prediction.

NeuralNoodles@CHiPSAL 2026: Late-Fusion Multimodal Stacking for Nepali Meme Sentiment Classification

Sidratul Muntaha[1], **Sabila Anzum**[1], **Arpita Mallik**[1], **Hasan Murad**[1]
[1]Chittagong University of Engineering & Technology (CUET), Chittagong, Bangladesh
{u2104119, u2104062, u2004023}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

Memes have emerged to be an essential medium of online expression, where the sentiment is determined by the interaction of text and image. Sentiment analysis of memes is particularly challenging when the language is low-resource, such as Nepali, due to the lack of resources and the complex relationships between text and image modalities. In this paper, we report our submission to Subtask B of CHiPSAL 2026, where the task was sentiment analysis of Nepali text-embedded memes for three sentiment classes: Negative, Neutral, and Positive. Through this submission, we present a late fusion multimodal framework that encompasses lexical, semantic, and visual models through a cross-validated stacking approach. Our submission to the shared task competition received a Macro F1 of 0.5045 on the official test set, achieving 6th place in the leaderboard. This demonstrates the strength of well-structured late fusion approaches to multimodal sentiment analysis of text-embedded memes.

Keywords: Multimodal Sentiment Analysis, Nepali Memes, Low-Resource NLP, Late Fusion, Meme Classification, Deep Learning

1. Introduction

Memes have become a popular way for people to communicate online. They allow individuals to express complex emotions, opinions, and viewpoints in an engaging and brief way Guleria et al. (2025). Automatic understanding of memes can give us valuable insights into public sentiment, cultural attitudes, and societal reactions to events Bucur et al. (2022). Unlike traditional text-only sentiment analysis, meme sentiment comes from the interplay between visual and textual elements, not from either part alone. This combination makes classifying meme sentiment a difficult research challenge.

While there has been significant progress in multimodal learning for English and other widely used languages, low-resource languages like Nepali are still largely unexplored. Identifying the sentiment in memes is important for understanding public opinion and for reducing the spread of negativity, misinformation, and harmful narratives Hossain et al. (2022).

Recent studies highlight the benefits of multimodal approaches. For example, Hossain et al. (2022) introduced MemoSen, a Bengali multimodal meme dataset and demonstrated that combining image and text features significantly improves classification performance over unimodal systems. Similarly, Jha et al. (2025) found that transformer-CNN hybrid architectures have also shown great performances in meme sentiment analysis and other related tasks. However, systematic approaches for multimodal sentiment analysis of Nepali memes are scarce, especially for structured fusion approaches that focus on robustness and generalization.

To fill this research gap, we have presented NeuralNoodlesCHiPSAL 2026, a late-fusion multimodal framework for Nepali meme sentiment analysis. The main contributions of this work are as follows:

1. To build a hybrid multimodal framework for Nepali meme sentiment analysis.
2. To design a structured probability-level late fusion strategy using cross-validated stacking.
3. To provide a reproducible and computationally efficient baseline for low-resource multimodal settings.

Although developing advanced cross-attention transformer models is beyond the scope of our proposed work, our proposed stacking-based multimodal model is a strong and reproducible baseline for future multimodal sentiment classification of Nepali memes. The implementation of our proposed method is publicly available at: `https://github.com/sidratul-m00ntaha/NeuralNoodles-CHiPSAL-2026`.

2. Related Work

Recent research has closely examined multimodal sentiment analysis of memes in different languages and tasks. Hossain et al. (2022) introduced a Bengali multimodal meme dataset with 4,368 annotated samples. They have showed that combining text and visuals improves performance by about 1.2% compared to unimodal baselines. Many transformer-based multimodal systems did very well on the Memotion shared tasks. Bucur et al. (2022) has used BERT, Sentence Transformers,

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 250–257
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

EfficientNet, and CLIP within a multi-task transformer framework, securing top ranks in sentiment and humor tasks. Similarly, Messina et al. (2021) proposed a Double Visual Textual Transformer (DVTT) architecture. This architecture uses mutually conditioned transformers for images and text, merging probability outputs by averaging. Sharma et al. (2020) has applied transfer learning with CNN backbones and stacked BiLSTM-GRU attention mechanisms for multimodal fusion. Swamy et al. (2020) has explored both unimodal and bimodal transformer-based architectures, pointing out challenges in effectively using visual cues.

Several studies have proposed fusion and hybrid architectures beyond shared tasks. Jha et al. (2025) evaluated DEIT+SBERT, DistillBERT+ResNet, RoBERTa+ResNet, and BiLSTM+ResNet, reporting 78.83% accuracy with DistillBERT+ResNet. Guleria et al. (2025) combined RoBERTa and CLIP for Hinglish meme sentiment classification, achieving 0.82 accuracy and significantly outperforming traditional machine learning methods. Vankov et al. (2025) created a dataset of 5,592 samples using distant supervision. They demonstrated that multimodal approaches and large language models do better than unimodal systems. Sharma et al. (2024) proposed ALFRED, an emotion-aware cross-modal gating framework that improved the F1-score by 4.94% over the baselines. In domain-specific detection tasks, Roy et al. (2024) used Vision Transformers and pretrained text transformers to detect Bangla cyberbullying memes, achieving a 0.76 F1-score. , Meanwhile, Shanmugavadivel et al. (2025) applied hybrid CNN-transformer architectures for detecting misogyny in Tamil memes. Hasan et al. (2022) explored deep learning techniques for detecting multimodal troll memes in Tamil. They focused on visual and textual feature extraction using VGG16, ResNet50, VGG19, CNN, and CNN+LSTM architectures. Their multimodal framework combined VGG16 image features with CNN and LSTM-based textual representations. The textual CNN+LSTM model achieved the highest weighted F1-score (0.52) and recall (0.57). The CNN-Text+VGG16 setup has reached the highest multimodal F1-score of 0.49, allowing the team to secure 4th place in the shared task. Velioglu and Rose (2020) used VisualBERT with ensemble learning to gain a 0.811 AUROC. Additionally, Elahi et al. (2023) approaches using ResNet50 and BanglishBERT showed promising results (0.71 weighted F1) for Bengali memes, highlighting the need for interpretability. Together, these studies show that multimodal transformer-based fusion, transfer learning, and ensemble strategies greatly improve meme classification performance. However, most research focuses on high-resource or better-resourced languages. Structured stacking-based late fusion strategies for Nepali meme sentiment analysis are still underexplored.

Motivated by existing multimodal sentiment analysis methods, we have proposed a hybrid late-fusion approach to handle sentiment classification in Nepali memes.

3. Dataset and Task

3.1. Shared Task Description

In this study, we have used the dataset provided in the Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes, held in CHiPSAL 2026 Thapa et al. (2026), where the focus is on the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) Sarveswaran et al. (2026). The shared task is about multimodal content moderation in low-resource languages, specifically Nepali memes.

The shared task is divided into two subtasks:

1. Hate Speech Detection in Nepali Only Memes

2. Sentiment Analysis in Nepali Only Memes

In this study, we have taken part in Subtask B, titled *Sentiment Analysis in Nepali Only Memes*, where the objective is to classify the memes according to the following sentiments: Negative, Neutral, and Positive.

3.2. Dataset Source and Construction

The dataset for this shared task was developed by extending the previous research works. To begin with, a multimodal framework for analyzing low-resource code-mixed memes was presented by Thapa et al. (2025a). This was followed by cross-platform retrieval and distillation techniques for improving the quality of the dataset, as presented by Thapa et al. (2025b). The annotation guidelines for detecting multimodal hate content were obtained from the CrisisHateMM framework, as presented by Bhandari et al. (2023)

3.3. Dataset Structure

The dataset is comprised of Nepali text-embedded meme image data. Table 1 presents the class-wise distribution of each set. Out of 1061 memes, 247 and 341 memes are from positive and negative classes respectively, while 473 are from the neutral class.

Class	Train	Test
Positive	247	–
Negative	341	–
Neutral	473	–
Total	1061	133

Table 1: Class-wise Distribution of the Dataset

4. Methodology

We have proposed a late-fusion multimodal framework for the sentiment classification of Nepali text-embedded memes (Subtask B). Due to the small dataset ($\sim$1k memes) and the complexity of multimodal reasoning, we have proposed a modular design that combines lexical, semantic, and visual representations using a stacking ensemble method.

Our work differs from other state-of-the-art transformer-based multimodal models such as the Double Visual Textual Transformer (DVTT) model proposed by Messina et al. (2021), the Visual BERT model used in Velioglu and Rose (2020) and the gated cross-attention model proposed by Sharma et al. (2024) in the sense that our model focuses on stability and generalization on a low-resource dataset. The overall architecture of the proposed framework is presented in Figure 1.

4.1. Problem Formulation

For a given meme

$$M = (I, T),$$

where I denotes the image and T denotes the OCR-extracted textual content, the goal is to predict a sentiment label:

$$y \in \{0, 1, 2\},$$

which represents:

- 0: Negative
- 1: Neutral
- 2: Positive

The training set has a moderate level of class imbalance, with the neutral class being the majority class. To evaluate the model on all classes equally, the model's performance has been evaluated using **Macro F1 score**, which gives equal importance to each class compared to accuracy score.

4.2. Data Preprocessing

Before feature extraction, some basic preprocessing steps were applied to the text as well as the image data.

For text:

- Removal of URLs and excessive whitespace
- Normalization of punctuation
- Lowercasing where appropriate
- Retention of Nepali Unicode characters

For images:

- All images were resized to 224 × 224 resolution
- Pixel normalization was applied using ImageNet statistics

These preprocessing steps ensure consistency across modalities while preserving sentiment-relevant features.

4.3. Textual Representation

For textual data representation, we have used two complementary representations:

4.3.1. Lexical features

We have used a hybrid representation, referred to as TF-IDF, that includes a bag of word features comprising word-level (1-2 gram) and character-level (3-5 gram) n-grams. The inclusion of character features is to account for variant spellings and informal language often utilized in memes. For classification, we have utilized a Logistic Regression model with balancing classes

4.3.2. Semantic features

We have used a multilingual Sentence Transformer model, referred to as paraphrase-multilingual-MiniLM-L12-v2, that produced embeddings of dimensionality 384 for each piece of text. The model was frozen to prevent potential overfitting. We chose it due to its good performance on multilingual data. Besides, its lightweight architecture makes it well-suited for low-resource settings compared to larger models such as XLM-RoBERTa or MuRIL. We have utilised a Logistic Regression classifier for this model. This dual modelling captured both surface-level sentiment cues and contextual semantics.

For the TF-IDF case, the word-level vectorizer had an n-gram range of (1,2) with a maximum vocabulary of 3000 features, while the character-level vectorizer had an n-gram range of (3,5) with a maximum of 2000 features. The final sparse representation was obtained by combining the word and character TF-IDF matrices, resulting in a feature space of up to 5000 dimensions. Logistic Regression was used with class balancing and a maximum of 1000 iterations.

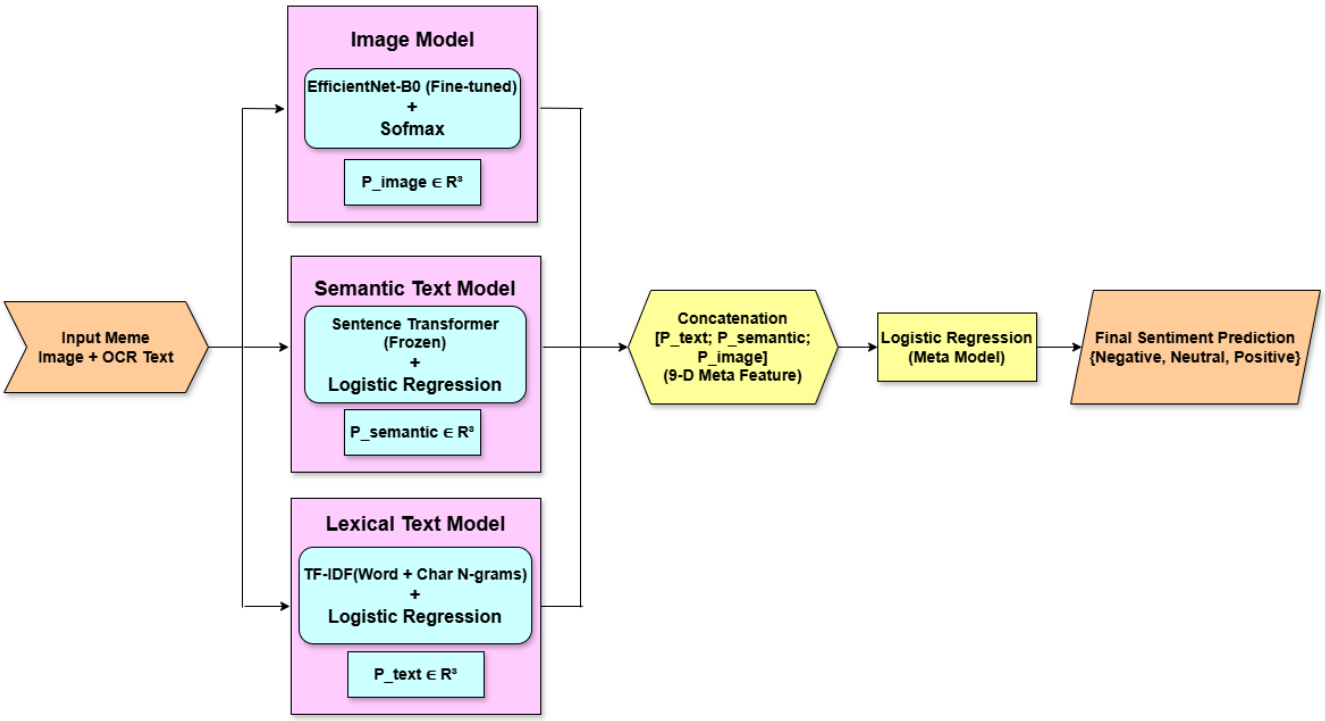


Figure 1: Methodological overview of the proposed method.

4.4. Visual Representation

To capture visual context such as facial expressions and symbolic imagery, we have used EfficientNet-B0 initialized with ImageNet-pretrained weights. The pooled feature vector of dimension 1280 is connected to the classification head.

The model was then fine-tuned using the AdamW optimizer, where the learning rate was set to 1e-4, and the batch size was set to 16. It was then trained for 4 epochs for each fold under stratified 5-fold cross-validation. No early stopping was employed, as the number of epochs for training was small.

The data augmentation techniques employed included resizing the images to 224x224, random horizontal flip, and random rotations ($\pm 10°$). For the validation, only image resizing and standardization of the mean and standard deviation of the ImageNet dataset were employed.

The backbone is fine-tuned using the cross-entropy loss function with moderate image augmentation, such as flipping the image horizontally and normalization. Fine-tuning enables adaptation to meme-specific visual patterns.

Prior work, including Bucur et al. (2022) and Roy et al. (2024), have shown that adapting visual backbones improves performance over unimodal approaches. However, we have used the lightweight CNN model rather than the Vision Transformers to maintain stability under limited data.

4.5. Late-Fusion Stacking Strategy

Each unimodal component (lexical text, semantic text, visual model) has produced a probability distribution over the three sentiment classes:

$$P_{\text{text}},\ P_{\text{semantic}},\ P_{\text{image}} \in \mathbb{R}^3$$

These probability vectors were concatenated to form a 9-dimensional meta-feature vector:

$$P_{\text{meta}} = [P_{\text{text}} \parallel P_{\text{semantic}} \parallel P_{\text{image}}]$$

A Logistic Regression meta-classifier was trained on these concatenated probabilities.

Although cross-modal transformers like VisualBERT make direct multimodal reasoning, they can be computationally costly and may suffer from overfitting, especially for small datasets. Inspired by ensemble strategies used in Hateful Memes Challenge in Velioglu and Rose (2020), we have used a late fusion approach for better robustness while maintaining modality-specific inductive biases.

We have used stratified 5-fold cross-validation to generate out-of-fold predictions for stacking, prevent information leakage and improve stability. Finally, predictions from each fold were averaged during the testing phase.

Overall, the multimodal sentiment classification model follows a design that favors modularity, interpretability, and generality.

5. Results and Analysis

5.1. Evaluation Metric

We have used **Macro-F1** to measure the performance of our system, which is also the official metric for the shared task. Macro-F1 assigns equal weight to each class, thus it is suitable for imbalanced sentiment distributions. We have also presented Precision, Recall, and F1 for individual classes to give a comprehensive view of the performance of our system. For all these metrics, we have used stratified 5-fold cross-validation to obtain the results.

5.2. Ablation Study on Modality Combinations

To understand the contribution of each modality in the system, we have conducted an ablation study on the performance of the unimodal models, the two modality combinations, and the complete three modality stacking framework. For this purpose, the Macro-F1 metric is used on the stratified 5-fold cross-validation protocol, and the results are reported in Table 2.

Model Configuration	Macro-F1
TF-IDF (Lexical)	0.3838
SentenceTransformer (Semantic)	0.3774
EfficientNet-B0 (Image)	0.3980
TF-IDF + Image	0.3878
SentenceTransformer + Image	0.3897
TF-IDF + SentenceTransformer	0.3715
TF-IDF + SentenceTransformer + Image	**0.4234**

Table 2: Macro-F1 scores of the unimodal models, the two modality combinations, and the complete three modality stacking framework.

In the case of unimodal models, the image model has the highest score, i.e., 0.3980, which indicates the importance of the visual modality in the sentiment classification of Nepali memes. The best-performing unimodal model i.e., the TF-IDF model, outperforms the semantic SentenceTransformer model, which indicates the importance of the surface-level textual features in the sentiment classification of the memes.

The use of two-modality combinations does not outperform the best unimodal model consistently. Although the TF-IDF + Image pair achieved a competitive result of 0.3878, as did the SentenceTransformer + Image pair with a result of 0.3897, these combinations did not outperform the image model alone. The textual combination of TF-IDF + SentenceTransformer, on the other hand, underperforms with a result of 0.3715, showing redundancy in the use of both textual representations.

On the contrary, the entire three-modality stacking model achieves the best Macro-F1 score of **0.4234**, which reflects a 6.4% improvement over the best unimodal model. This suggests that the use of lexical, semantic, and visual representations offers complementary information, and the use of the three modalities altogether is a prerequisite to improve the sentiment classification of Nepali memes using the multimodal approach.

It is important to note that there is a notable difference between the cross-validation Macro F1 score and the test set Macro F1 score (0.4234 and 0.5045, respectively). This can be explained by the fact that the data may not be evenly distributed in the training and test datasets, and the dataset is relatively small. Since our approach relies on stratified 5-fold cross-validation without post-hoc tuning on the test set, the cross-validation score provides a conservative estimate of model performance, while the higher test score reflects better generalisation on the unseen data.

5.3. Cross-Validation Stability

To measure model's robustness, the Macro-F1 score is provided fold-wise, as shown in Table 3.

Fold	Macro-F1
1	0.419271
2	0.421013
3	0.454797
4	0.407765
5	0.407215

Table 3: Fold-wise Macro-F1 scores

The performance ranges from 0.4072 to 0.4548, with a mean of 0.4220 and a variation range of 0.0476. This shows that the model's performance is quite stable, implying that the stacking model reduces overfitting.

5.4. Per-Class Performance

Table 4 presents class-wise evaluation metrics.

Class	Precision	Recall	F1
Negative	0.4447	0.5777	0.5026
Neutral	0.5375	0.3636	0.4338
Positive	0.3054	0.3684	0.3339

Table 4: Per-class performance of the stacked model

The model performs best on the *Negative* class (F1 = 0.5026). This is because the model's recall is quite high. This shows that the model is good at identifying explicit negative sentiment signals. Performance on the *Neutral* class is moderate with

relatively low recall(F1 = 0.4338). This shows that the model is not good at identifying the class of ambiguous text. The *Positive* class yields the lowest F1 score (0.3339), which shows that the model is not good at distinguishing positive sentiment from neutral or sarcastic content.

5.5. Confusion Matrix Analysis

Figure 2 illustrates the confusion matrix of the final stacked model.

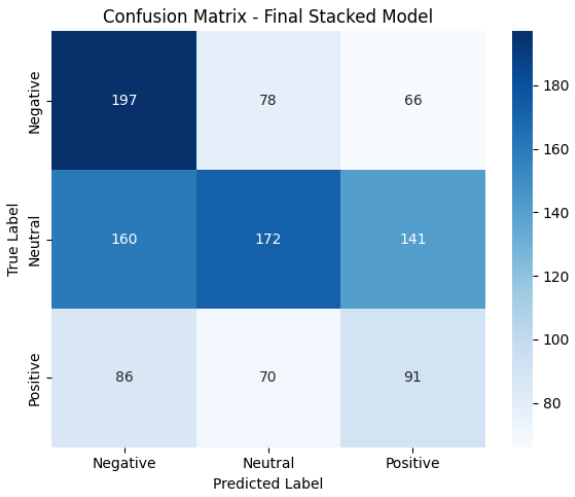


Figure 2: Confusion matrix of the stacked multimodal model.

The confusion matrix shows that there is a significant overlap between the Neutral class and the other two classes. There is a significant number of Neutral instances that have been incorrectly predicted to be Negative and Positive. This shows that the semantic complexity and ambiguity of meme content make sentiment analysis difficult. The Negative class has the highest number of correct predictions, showing that sentiment detection is relatively stable for this class. In contrast, the Positive class is frequently misclassified as Negative, suggesting that subtle positive sentiment and sarcasm in meme content are difficult to detect.

Overall, the proposed late fusion multimodal framework outperforms all the baseline models, and the performance is stable across all cross-validation models. The results confirm that integrating lexical, semantic, and visual representations enhances sentiment classification in low-resource meme settings. However, the moderate Macro F1 score and poor performance of the model on the Positive class show that sentiment understanding in multimodal memes is difficult.

6. Conclusion

In this paper, we have proposed a late fusion multimodal approach for sentiment classification of Nepali text-embedded memes in a low-resource scenario. The proposed method incorporates lexical TF-IDF features, semantic sentence embeddings and fine-tuned image features via a stacking-based meta-classifier to achieve a Macro F1 score of 0.4234 in cross-validation, which surpasses unimodal baselines. On the official test set, the model achieved a higher Macro-F1 score of 0.5045, securing 6th position, which indicates strong generalisation despite the limited training data. The image-based model exhibits the best performance individually, indicating the importance of image content in meme sentiment classification. The stacking-based fusion effectively combines modality-specific features while maintaining consistent performance across different folds in cross-validation. The overall performance is moderate, while there are challenges in classifying the Positive class. The proposed method can be a starting point to explore better cross-modality interaction mechanisms to improve sentiment classification. The overall study can serve as a reproducible and modular baseline for multimodal sentiment classification in low-resource meme datasets.

Overall, this study provides a reproducible and modular baseline for multimodal sentiment classification in low-resource meme datasets.

7. Limitations

Although the suggested late fusion multimodal model presents competitive results, there are a number of limitations that need to be noted. Firstly, the dataset size is not large, as there are only 1,061 training samples. This makes it difficult for deep models to fully grasp the complex relationships between the modalities. The model also seems to have difficulty in recognizing Positive sentiment; this could be due to difficulties in recognizing sarcasm or irony.

Secondly, the suggested model relies on text features obtained through OCR from the image dataset, which may not capture culturally specific features that may be part of the meme. Finally, the class imbalance that may affect the detection of certain classes. In future work, there may be a number of directions that can be explored, including data augmentation, class imbalance, and cross-modal interaction.

8. Ethics Statement

This study undertook a multimodal sentiment analysis of Nepali memes and this was conducted within

a shared-task framework. The dataset was made available by the organizers. No personal data was collected for this study.

It is to be noted that, although sentiment analysis tools are helpful for understanding content and promoting digital well-being, automated models may also contain biases, which are usually reflected by the data on which these models are trained. These biases may result in misclassifications, particularly for ambiguous and sarcastic memes, which are common in social media.

Thus, we would like to emphasize that, multimodal sentiment models should be used responsibly and human intervention should also be considered to ensure fairness and context-awareness, particularly for such models that may contain biases.

9. References

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Ana-Maria Bucur, Adrian Cosma, and Ioan-Bogdan Iordache. 2022. Blue at memotion 2.0 2022: You have my image, my text and my transformer.

Kazi Toufique Elahi, Tasnuva Binte Rahman, Shakil Shahriar, Samir Sarker, Sajib Kumar Saha Joy, and Faisal Muhammad Shah. 2023. Explainable multimodal sentiment analysis on bengali memes.

A. Guleria, K. Varshney, G. Pahwa, et al. 2025. Multimodal sentiment analysis of english and hinglish memes. *Multimedia Tools and Applications*, 84:15331–15356.

Md Hasan, Nusratul Jannat, Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. CUET-NLP@DravidianLangTech-ACL2022: Investigating deep learning techniques to detect multimodal troll memes. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 170–176, Dublin, Ireland. Association for Computational Linguistics.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. MemoSen: A multimodal dataset for sentiment analysis of memes. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1542–1554, Marseille, France. European Language Resources Association.

Rishav Jha, Mohit Ranjan Panda, Shubham K C, and Aaditya Dahal. 2025. Deep learning architectures for multimodal sentiment analysis. In *2025 International Conference on Intelligent and Cloud Computing (ICoICC)*, pages 1–6.

Nicola Messina, Fabrizio Falchi, Claudio Gennaro, and Giuseppe Amato. 2021. AIMH at SemEval-2021 task 6: Multimodal classification using an ensemble of transformer models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1020–1026, Online. Association for Computational Linguistics.

Aishwaria Roy, Hasan Murad, and Ayman Iktidar. 2024. A multimodal approach to bangla cyberbullying meme detection in social media. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 1886–1891.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Kogilavani Shanmugavadivel, Malliga Subramanian, Mohamed Arsath H, Ramya K, and Ragav R. 2025. TEAM_STRIKERS@DravidianLangTech2025: Misogyny meme detection in Tamil using multimodal deep learning. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 619–623, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Mayukh Sharma, Ilanthenral Kandasamy, and W.b. Vasantha. 2020. Memebusters at SemEval-2020 task 8: Feature fusion model for sentiment analysis on memes using transfer learning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1163–1171, Barcelona (online). International Committee for Computational Linguistics.

Shivam Sharma, Ramaneswaran S, Md. Shad Akhtar, and Tanmoy Chakraborty. 2024. Emotion-aware multimodal fusion for meme emotion detection. *IEEE Transactions on Affective Computing*, 15(3):1800–1811.

Steve Durairaj Swamy, Shubham Laddha, Basil Abdussalam, Debayan Datta, and Anupam Jamatia. 2020. NIT-agartala-NLP-team at SemEval-2020 task 8: Building multimodal classifiers to tackle Internet humor. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1179–1189, Barcelona (online). International Committee for Computational Linguistics.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Georgi Vankov, Dimitar Dimitrov, Ivan Koychev, and Preslav Nakov. 2025. Multimodal sentiment analysis: Recognizing sentiment in memes. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 1–11, Cham. Springer Nature Switzerland.

Riza Velioglu and Jewgeni Rose. 2020. Detecting hate speech in memes using multimodal deep learning approaches: Prize-winning solution to hateful memes challenge.

Multi-Modal-Minds@CHiPSAL 2026: A Comparative Study of Textual, Visual and Multimodal Architecture for Nepali Meme Moderation

Sandesh Shrestha[1], **Bikram K.C.**[1], **Akshyat Shah**[2], **Ashish Acharya**[3], **Rabin Thapa**[1]
[1] IIMS College, Kathmandu, Nepal
{sandeshrestha115, bikramkc9810}@gmail.com, rabin@iimscollege.edu.np
[2] Delhi Technological University, Delhi, India
akshyatshah@gmail.com
[3] Kathmandu University, Dhulikhel, Nepal
ashishacharya@ku.edu.np

Abstract

Memes have become ubiquitous on social media platforms blending text and imagery to express complex and culturally nuanced messages. While a high degree of automation in meme moderation has been achieved for high-resource languages, low-resource languages, such as Nepali, still remain largely neglected. In this paper, we describe our system submission to the CHiPSAL 2026 Shared Task on Multi-modal Hate and Sentiment Understanding in Low-Resource Nepali Memes, which features two main sub-tasks: (1) Detection of Hate Speech as binary classification and (2) Sentiment Analysis as multi-class classification in Nepali memes. We perform a comprehensive analysis of the following models: uni-modal textual models (mBERT, XLM-RoBERTa, MuRIL), uni-modal visual models (ResNet, ConvNeXt, ViT), nine different early-fusion multimodal models, and the vision-language foundation model, SigLIP. Among all models, the ViT model achieved the best macro F1-score (0.6278) for the hate speech detection task, while SigLIP achieved the best score (0.5481) for the sentiment analysis task. We hypothesize that the under-performance of fusion models may be attributed to OCR noise and inadequate low-resource textual representations that act as a bottleneck when paired with more advanced visual encoders. These results highlight the unique challenges of multimodal meme comprehension in low-resource contexts and underscores the requirement for culturally grounded, noise-robust approaches to content moderation in Nepali.

Keywords: Nepali memes, Hate Speech Detection, Sentiment Analysis, Multi-modal Learning, Low-Resource NLP

1. Introduction

There are numerous social networks that have significantly changed the way varying sentiments, humor, and opinions are disseminated. With memes becoming one of the most popular formats for information exchange in contemporary society, the dynamics of social discourse have also become more multifaceted. Unlike traditional approaches that use text for expression, memes combine textual and visual media to convey nuanced meaning, often relying on subtle cultural contexts (Kiela et al., 2020a). Although many memes are humorous or benign, a significant subset tends to skew towards hate speech or invoke sentiment that may influence social attitudes. Automatic identification of harmful or sentiment-laden memes is a growing area of research within natural language processing (NLP) and computer vision.

Detecting hate speech and classifying sentiment in memes is particularly tedious and challenging due to multi-modal interactions as neither text nor image alone captures the full meaning(Parihar et al., 2021). Most of the existing work on ML and language processing is specifically limited to high-resource languages such as English, while low-resource languages such as Nepali remain underserved and under-explored (Thapa et al., 2025a). The lack of sufficient data and pre-trained models to support robust multi-modal understanding in under-resourced languages is a pressing need. Thus, research that organically focuses on such languages is essential for equitable and worldwide content moderation.

The Shared Task (Thapa et al., 2026) on Multi-modal Hate and Sentiment Understanding in Low-Resource Memes at CHiPSAL 2026 (Sarveswaran et al., 2026) provides a benchmark for addressing this problem specifically in Nepali. The task uses a dataset curated from (Thapa et al., 2025a) and contains Nepali-only memes annotated for hate speech and sentiment, and challenges participants to develop capable systems for binary hate detection and three-class sentiment classification (negative, neutral, positive).

2. Related Works

Although substantial work has been done on hate speech and sentiment analysis in English, non-English languages particularly low resource ones have remained largely underrepresented in literature. We therefore focus on extensions into the Nepali language, as limited work has been done in

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 258–266
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

hate speech detection and sentiment analysis.

In their work, (Kiela et al., 2020a) introduce the Hateful Memes Challenge, a large-scale benchmark of multimodal memes. The paper shows that text-only models marginally outperform vision-only models. They observe that the more advanced the fusion, the better the model performs. The paper suggests that early fusion models (MMBT, ViLBERT and Visual BERT) broadly outperform middle (Concat) and late fusion approaches with ViLBERT CC giving the highest accuracy of 66.10 among all the methods. (Bhandari et al., 2023) introduce CrisisHateMM, a multimodal dataset of over 4,700 text-embedded images collected from the Russia-Ukraine conflict and annotated for directed and undirected hate speech using a CLIP-based multimodal fusion architecture. Their best-performing model achieves F1-scores of 0.786, 0.609, and 0.615 for hate speech, its direction and targets respectively, demonstrating that combining visual and textual modalities significantly outperforms text-only and image-only baselines.

(Chakravarthi et al., 2020, 2021) address the lack of resources for Dravidian languages by constructing an annotated corpus of 15,744 sentences and organizing shared tasks on sentiment analysis and offensive language detection for code-mixed social media text. They released datasets comprising 43,919 Tamil, 20,010 Malayalam, and 7,772 Kannada comments collected from YouTube. The highest-performing system used an ensemble of transformer models selected via a genetic algorithm, achieving weighted F1-scores of 0.78 for Tamil, 0.97 for Malayalam, and 0.75 for Kannada through task-adaptive pretraining of mBERT and XLM-RoBERTa (Sitaula et al., 2021).

(Sitaula et al., 2021) presented one of the first large-scale benchmarks for Nepali social media sentiment analysis, focusing on 33,247 COVID-19-related tweets. Their experiments with deep learning models, including CNNs, yielded a best overall classification accuracy of 68.7

(Thapa et al., 2023) introduced NEHATE, a manually annotated dataset of 13,505 Nepali tweets for hate speech and target identification in local election discourse. Using multilingual BERT variants, NepBERTa achieved the best performance with macro F1-scores of 0.68 for hate speech detection and 0.60 for target identification, establishing NEHATE as one of the first large-scale hate speech resources for Nepali. (Thapa et al., 2025a) propose a multi-modal prompt-based framework evaluated on low resource Nepali and code-mixed Nepali meme datasets. By using their proposed MemeNePAL model, they achieved an F1-score of 0.5301 and 0.6331 across sentiment analysis and hate detection respectively, higher compared to other text-only, vision-only and multi-modal models. The paper also introduces NeMeme, the first Nepali meme dataset annotated for hate speech and sentiment analysis. (Thapa et al., 2025b) present a cross-platform retrieval-augmented distillation approach for 4,211 code-switched memes in the Nepali-English language classified for sentiment and hate speech. They propose MM-RAD model for code switched context achieving an F1-score of 43.24 in sentiment analysis and 64.96 in hate speech detection.

Building on the dataset published in (Thapa et al., 2025a), the shared task (Thapa et al., 2026) at CHiPSAL@LREC 2026 (Sarveswaran et al., 2026) extends the challenge to the community. They invite methods for multi-modal hate and sentiment understanding on low-resource Nepali memes serving as the basis for this paper.

3. Dataset and Task

This work is part of the Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes (Thapa et al., 2026) at CHiPSAL 2026 (Sarveswaran et al., 2026). The shared task consists of two different subtasks: Subtask A aims to detect hate speech via binary classification into hate and non-hate while Subtask B is associated with the three way classification of sentiment into positive, negative and neutral classes. The shared task is hosted on Codabench and a dataset for this task has already been published in ICWSM 2025 (Thapa et al., 2025a) as a part of NeMeme. The dataset contains memes collected from social media platforms, and textual content embedded within meme images is extracted using EasyOCR with Nepali language support. However, EasyOCR's performance on informal, colloquial Devanagari script as commonly found in internet memes is known to be inconsistent, often introducing transcription noise that can degrade downstream textual representations. Table 1 summarizes the distribution between all splits for the shared task.

The relatively small training set of approximately 1,000 samples per subtask poses a particular challenge for multimodal fusion models, which introduce significantly more parameters than their unimodal counterparts. Learning effective cross-modal alignment under such data constraints increases the risk of overfitting, which may partially explain the comparatively weaker performance of explicit fusion architectures observed in our experiments.

3.1. Subtask A: Hate Speech Detection

Subtask A requires binary classification of Nepali-only memes into hate and non-hate categories. The training set comprises 1,068 memes, of which

Subtask	Label	Train	Val	Test
A (Hate Detection)	Hate	348	35	–
	Non-Hate	720	98	–
	Total	1,068	133	134
B (Sentiment Analysis)	Negative	341	39	–
	Neutral	473	65	–
	Positive	247	29	–
	Total	1,061	133	133
Total		**2,129**	**266**	**267**

Table 1: Distribution across train, validation, and test splits for Subtask A (Hate Detection) and Subtask B (Sentiment Analysis) of the CHiPSAL 2026 shared task (Thapa et al., 2026). Test set class-level labels were withheld by the shared task organizers; only the total instance count is available.

720 are Non-Hate(labeled as 0) and 348 are Hate(labeled as 1). The validation set consists of 133 memes with 98 Non-Hate and 35 Hate instances. The test set, on the other hand, contains 134 unlabeled memes for the final evaluation. Figure 1 gives an idea of the instances in the Hate Detection subtask.



(a) Hate



(b) Non-Hate

Figure 1: Examples of Nepali memes from the shared task (Thapa et al., 2026) for Subtask A (Hate Speech Detection).

3.2. Subtask B: Sentiment Analysis

Subtask B requires a multi-class sentiment classification of Nepali-only memes into Negative(labeled as 0), Neutral(labeled as 1), and Positive(labeled as 2) categories. The training set contains 1,061 memes with 341 negative, 473 neutral and 247 positive text embedded images. The validation set consists of 133 memes with 39 negative, 65 neutral, and 29 positive instances. The test set contains 133 unlabeled memes for the final evaluation. Figure 2 gives an idea of the instances in the Sentiment Analysis subtask.



(a) Negative



(b) Neutral



(c) Positive

Figure 2: Examples of Nepali memes from the shared task (Thapa et al., 2026) for Subtask B (Sentiment Analysis).

4. Methodology

We developed a comprehensive experimental framework to process the raw dataset and establish robust unimodal baselines. Using these baselines, we evaluated two multimodal paradigms: an exhaustive grid of nine early-fusion architectures and a pre-aligned vision-language foundation model.

4.1. Textual & Visual Prepossessing

The EasyOCR framework, configured with Devanagari and English language support, was utilized to extract text from the meme images. To evaluate the models under realistic, "in-the-wild" moderation conditions, we applied EasyOCR directly to the raw images without explicit image-level text preprocessing (such as binarization, deskewing, or contrast adjustment). While this preserves the original structure of the memes, the highly variable nature of meme typography (e.g., erratic stroke widths, dynamic colors, and chaotic backgrounds) introduces an inherent baseline of OCR noise. The raw strings resulting from this extraction were then passed directly to our subword tokenizers to accommodate unknown vocabulary items and local slang.

We built a stochastic enhancement pipeline for our visual modality through the application of the *albumentations* library (Buslaev et al., 2020), which enabled our encoders to acquire strong semantic characteristics that protected it from learning specific patterns of internet memes. The images were resized and scaled to 224×224 pixels. The model was trained using the following data augmentation pipeline:

- **Web Degradation:** We simulate the characteristic artifact of the platform using two methods, which include JPEG compression at quality

levels between 30 and 70 and aggressive spatial down-scaling using scale factors between 0.5 and 0.8, with a probability of 40%.

- **Spatial Regularization:** We use coarse dropout (cutout) to up to four rectangular regions with a 30% probability, preventing the model from over-relying on isolated visual cues.

- **Color Variation:** We dynamically apply random hue and saturation adjustments with the probability of 30% which prevents the model from over-relying on isolated visual cues.

Finally, all augmented image tensors are normalized using standard ImageNet statistics before being passed to the visual encoders.

4.2. Experimental Setup

Hyperparameter	Value
Optimizer	AdamW ($\epsilon = 1 \times 10^{-8}$)
β_1, β_2	0.9, 0.999
Batch Size	16
Dropout (Head)	0.3
Epochs (Max)	15
Early Stopping	Patience = 5
Hardware	NVIDIA Tesla P100
Random Seed	42

Table 2: General training configurations applied across all experiments.

A consistent training pipeline was employed across both the Hate Speech Detection and Sentiment Analysis subtasks The CHIPSAL 2026 Nepali meme dataset (Thapa et al., 2026) served as the primary benchmark for all model evaluations

The preprocessed inputs, as detailed in the previous section, were utilized to train the various architectures Specifically, the textual modality relies on OCR-extracted Nepali text, while the visual modality leverages stochastically augmented meme images This standardized setup ensured a fair and unbiased evaluation across all unimodal and early-fusion multimodal systems

As summarized in Table 2, key hyperparameters included a batch size of 16, the AdamW optimizer, and a maximum of 15 training epochs with early stopping (patience = 5) Parameter-efficient fine-tuning was achieved using LoRA ($r = 16$, $\alpha = 32$) for both text and image encoders A complete list of model-specific learning rates, is provided in Appendix 9 for full reproducibility

4.2.1. Unimodal and Multimodal Models Selection

Unimodal Textual Models: We evaluated multiple text-only transformer-based models for Nepali meme analysis. As justified above, we utilized mBERT-base-uncased (Devlin et al., 2019), XLM-RoBERTa-base (Conneau et al., 2020), and MuRIL (Khanuja et al., 2021). From these models, we extracted independent textual embeddings $\mathbf{E}_{\text{text}} \in \mathbb{R}^{d_t}$ to serve as our baseline textual framework.

Uni-modal Visual Models: We extracted visual embeddings $\mathbf{E}_{\text{image}} \in \mathbb{R}^{d_v}$ using a diverse set of advanced visual architectures: ResNet (He et al., 2016), ConvNeXt (Liu et al., 2022), and Vision Transformer (ViT) (Dosovitskiy et al., 2021).

Multi-modal Models: To capture cross-modal interactions, we experimented with early-fusion architectures by pairing each visual encoder with each textual encoder. Specifically, we combined the set of visual models $\{\text{ResNet}, \text{ConvNeXt}, \text{ViT}\}$ with the set of textual models $\{\text{mBERT}, \text{XLM-R}, \text{MuRIL}\}$, resulting in a total of nine multimodal configurations.

Following standard early-fusion (feature-level) paradigms for multi-modal meme classification (Baltrušaitis et al., 2018; Kiela et al., 2020b), textual and visual representations are extracted independently before being combined for classification. Formally, given an image-text pair (x_v, x_t), let f_v and f_t denote the visual and textual encoders, respectively. We first extract the independent dense feature vectors $\mathbf{h}_v = f_v(x_v)$ and $\mathbf{h}_t = f_t(x_t)$. These representations are then concatenated to form a fused multimodal vector:

$$\mathbf{h}_{fused} = [\mathbf{h}_v \oplus \mathbf{h}_t]$$

where $\oplus$ denotes the concatenation operation. Finally, this joint representation is passed through a linear classification head to obtain the predicted probabilities $\hat{y}$:

$$\hat{y} = \text{Softmax}(\mathbf{W}\mathbf{h}_{fused} + \mathbf{b})$$

where $\mathbf{W}$ and $\mathbf{b}$ represent the learnable weights and bias of the classifier. This constitutes an early-fusion paradigm, as the representations from both modalities are merged prior to the final classification layer, in contrast to decision-level late fusion where independent model predictions are combined.

Vision-Language Model: In addition to encoder-based fusion approaches, we also evaluated SigLIP (Zhai et al., 2023), a pretrained vision-language model that jointly learns aligned representations between images and text.

4.2.2. Architectural Justification:

The selection of our diverse model architectures is deeply grounded in the linguistic and visual realities of Nepali digital culture, as highlighted in recent

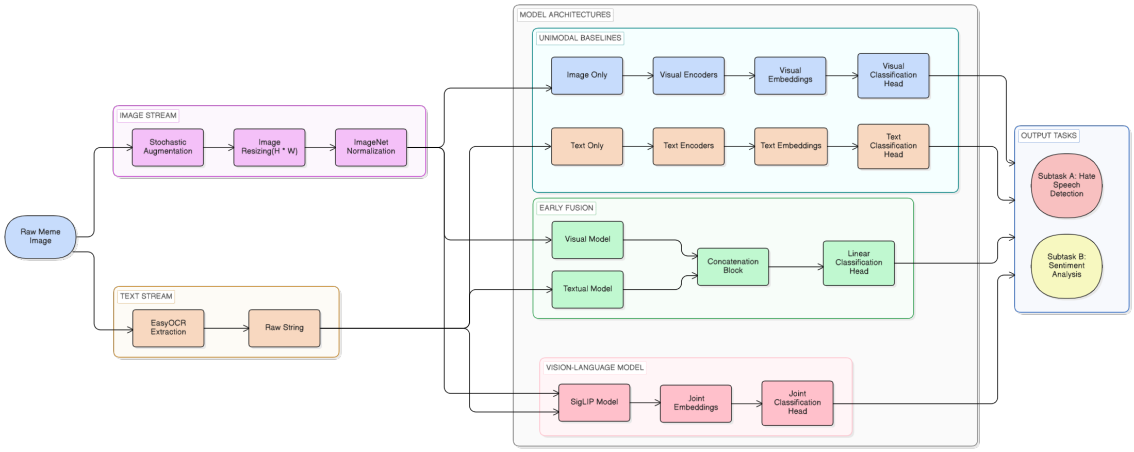


Figure 3: Unified system architecture illustrating the experimental framework. The raw meme input is split into independent preprocessing streams: the Image Stream utilizes stochastic augmentation to prevent pattern memorization, while the Text Stream extracts raw Devanagari strings via EasyOCR. The framework evaluates three distinct modeling paradigms—Unimodal Baselines, explicit Early Fusion networks, and a pre-aligned Vision-Language Model (SigLIP)—across the final output tasks of Hate Speech Detection and Sentiment Analysis.

multimodal literature (Kiela et al., 2020b; Bhandari et al., 2023). For the textual modality, we deliberately prioritized multilingual models (mBERT, XLM-ROBERTa) and specifically MuRIL, rather than relying solely on monolingual Nepali encoders. Nepali memes frequently exhibit complex code-switching not only with English but heavily with Hindi, driven by the pervasive influence of Bollywood and the Indian entertainment industry on regional internet culture (Thapa et al., 2025a). MuRIL's specific pre-training on Indic languages makes it uniquely equipped to handle this nuanced, cross-lingual meme vocabulary. For the visual modality, we selected ResNet to serve as a proven, robust baseline, ConvNeXt to evaluate modern CNN optimizations, and the Vision Transformer (ViT) to capture the global context of visual semiotics (e.g., emojis and specific meme templates). Finally, our inclusion of the SigLIP foundation model was driven by the proven efficacy of pre-aligned cross-modal representations in noisy, low-resource moderation tasks.

5. Results & Discussion

Table 3 presents a comparative performance analysis of the evaluated architectures across the Hate Speech Detection and Sentiment Classification subtasks.

5.1. Hate Speech Detection:

Transformer-based visual architectures proved highly effective for the hate speech detection task. While the prevailing narrative suggests that multimodal fusion is inherently superior for meme moderation, our standalone Vision Transformer (ViT) achieved the highest overall Macro F1-score of 0.6278, driven by the highest recall (0.6278) among all evaluated models.

Equally notable is the performance of the vision-language foundation model, SigLIP. While its F1-score of 0.5999 slightly trailed the pure ViT, SigLIP achieved the highest overall accuracy (0.6940) and a strong precision of 0.6424. This demonstrates that SigLIP's pre-trained cross-modal alignment makes it highly reliable for general classification across both classes, even if the pure visual signal of the ViT was slightly more effective at retrieving specific hateful instances.

Further examination of the metrics revealed distinct behavioral differences across the models. The text-only baseline, XLM-RoBERTa, achieved a high precision of 0.8383 but an extremely low recall of 0.5114. This indicates that while XLM-RoBERTa rarely misclassified regular memes as hateful, it overlooked a considerable amount of genuinely hateful content. Explicit multi-modal architectures, such as MuRIL-ConvNeXt (F1-score of 0.6072), performed competitively but did not surpass the performance of the pure visual signal represented by the ViT.

5.2. Sentiment Analysis:

A similar reliance on pre-trained visual alignment for sentiment classification was observed in Subtask B. The SigLIP model achieved the best performance, sweeping all evaluation metrics with the highest F1-score of 0.5481, a high precision of 0.5067, a recall of 0.6091 and an outstanding accuracy of

	Models	Hate Detection				Sentiment Analysis			
		Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.
Text	mBERT	0.5322	0.5318	0.5320	0.5896	0.4254	0.4208	0.3831	0.3985
	XLM-RoBERTa	**0.8383**	0.5114	0.4258	0.6791	0.1579	0.3333	0.2143	0.4737
	MuRIL	0.3358	0.5000	0.4018	0.6716	0.4254	0.4208	0.3831	0.3985
Visual	ResNet	0.6178	0.5811	0.5804	0.6791	0.4955	0.4753	0.4793	0.5188
	ConvNeXt	0.5872	0.5763	0.5782	0.6493	0.3905	0.3985	0.3897	0.4511
	ViT	0.6278	**0.6278**	**0.6278**	0.6716	0.4140	0.4104	0.4112	0.4436
Explicit Fusion	mBERT + ResNet	0.5890	0.5944	0.5904	0.6269	0.3051	0.3049	0.2840	0.2857
	mBERT + ConvNeXt	0.5795	0.5902	0.5682	0.5821	0.4083	0.4123	0.4076	0.4211
	mBERT + ViT	0.5916	0.6005	0.5915	0.6194	0.4828	0.4984	0.4826	0.4887
	MuRIL + ResNet	0.5601	0.5601	0.5601	0.6119	0.2634	0.2579	0.2402	0.2406
	MuRIL + ConvNeXt	0.6101	0.6235	0.6072	0.6269	0.4320	0.4362	0.4276	0.4436
	MuRIL + ViT	0.5774	0.5833	0.5780	0.6119	0.4713	0.4657	0.4677	0.4887
	XLM-RoBERTa + ResNet	0.6014	0.6119	0.6012	0.6269	0.1322	0.3083	0.1682	0.2180
	XLM-RoBERTa + ConvNeXt	0.5589	0.5710	0.5598	0.5675	0.3698	0.3698	0.3685	0.3985
	XLM-RoBERTa + ViT	0.5352	0.5384	0.5335	0.5672	0.4561	0.4562	0.4491	0.4662
V-L	**SigLIP**	0.6424	0.5980	0.5999	**0.6940**	**0.5067**	**0.6091**	**0.5481**	**0.7143**

Table 3: Performance comparison of unimodal and multimodal architectures for sentiment and hate analysis. Bold values denote the best-performing model for each metric within a given subtask.

0.7143.

On the other hand, unimodal text models underperformed on this subtle task. For instance, XLM-RoBERTa struggled significantly, achieving a precision of only 0.1579. Although explicit fusion methods such as mBERT-ViT achieved a reasonable F1 score of 0.4826 and an accuracy of 0.4887, it was significantly lower than the SigLIP baseline. Since SigLIP is already trained on a high-quality, cross-learning image-text alignment, its performance implies that the use of a foundation model with acquired cross-modal knowledge is much more effective than directly combining low-resource Nepali text encoders with standard vision models.

5.3. Discussion of Findings

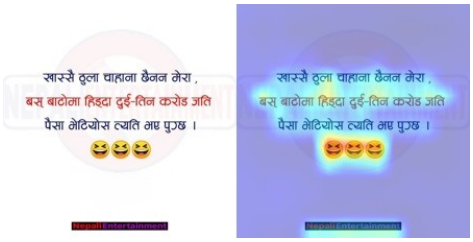
The primary finding across both subtasks is that unimodal visual architectures and pre-aligned vision-language foundation models unexpectedly outperformed explicit early-fusion integration networks. While prevailing assumptions in meme moderation suggest that multimodal fusion is inherently superior, our empirical results demonstrate that the visual modality carries the primary semantic burden in this specific low-resource, high-noise environment.

We postulate that this performance gap is directly attributable to the "distractor" effect caused by degraded textual signals. In high-noise environments characterized by complex code-switching, casual slang, and intentional visual degradation (such as "deep-fried" aesthetics), the OCR extraction pipeline frequently produces noisy or incoherent text inputs. When these degraded textual representations are concatenated with strong visual features during early fusion, the textual modality acts as an active distractor rather than a complementary signal. The linear classification head struggles to isolate the predictive visual signal from the high-entropy textual noise, creating a significant performance bottleneck.

Consequently, standalone visual models like the Vision Transformer (ViT) perform better precisely because they are insulated from this textual noise. By treating typographical overlays and emojis strictly as visual semiotics, the ViT avoids being confused by OCR transcription errors. Similarly, foundation models like SigLIP, which are pre-trained on massive datasets to align images and text holistically, demonstrate a superior ability to filter out this modality-specific noise compared to fusion networks trained from scratch on our limited dataset. Ultimately, these findings underscore that forcing multimodal fusion with unreliable text inputs is actively detrimental, and robust moderation in low-resource languages must prioritize noise-resilient or entirely vision-dominant approaches.

6. Interpretability and Modality Reliance

We conducted a two-fold interpretability analysis: (1) mapping spatial attention on the unimodal Vision Transformer (ViT) using Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize modality-specific focus areas, and (2) performing



(a) Correct Classification (True: Hate, Pred: Hate)



(b) Incorrect Classification (True: Hate, Pred: Non-Hate)

Figure 4: Grad-CAM heatmaps from the penultimate layer of the ViT model. Subfigure (a) correctly identifies hateful intent by focusing on visual semiotics (emojis), while (b) shows a failure to capture context, misidentifying a hateful meme as Non-Hate

counterfactual modality ablation using the vision-language foundation model (SigLIP).

6.1. Visual Attention and Semiotics (ViT)

To visualize how the text-agnostic ViT handles multimodal inputs, we generated Grad-CAM heatmaps from the penultimate layer of the model. Because ViT lacks a natural language tokenizer, it processes typographical overlays strictly as visual tokens.

As shown in Figure 4 (a), a hateful meme was successfully classified. The activation map shows that the model is largely invariant to the structural Devanagari text; instead, it centers its attention on the laughing emojis (specifically, the *face with tears of joy* emoji). Emojis play a significant role and can be regarded as strong visual semiotics that carry significant emotive weight in meme culture. In most cases, emojis express mockery, sarcasm, or malevolent intent. Standard low-resource NLP pipelines often do not recognize the semantic content of such Unicode characters. Nevertheless, these are leveraged effectively by the ViT as highly discriminative visual characteristics, hence its superior F1-score in hate speech classification.

On the other hand, Figure 4 (b) shows a critical failure mode (True: 1, Predicted: 0). The Grad-CAM heatmap shows that the model focuses disproportionately on a visually secondary object (a monkey). The model fails to capture the contextual hatefulness created by the interaction of the text and image because it lacks the ability to process the accompanying text. This proves that while visual semiotics are powerful, this reliance creates significant blindspots regarding context-specific memes where the image alone does not convey the full intent.

Modality Reliance	Hate Speech	Sentiment
Image Bias	34 (25.6%)	12 (9.0%)
Text Bias	6 (4.5%)	17 (12.8%)
True Multimodal	93 (69.9%)	104 (78.2%)
Total	**133**	**133**

Table 4: Counterfactual modality ablation results for the SigLIP model, detailing unimodal bias versus true cross-modal synergy across both subtasks.

6.2. Counterfactual Modality Ablation (SigLIP)

We employed a counterfactual ablation framework to evaluate modality bias and inter-modality dynamics in the SigLIP architecture. By applying both text and image ablation to each validation sample, we isolated the modality driving the baseline multimodal prediction. An instance was defined as having an image or text bias if only one of the unimodal predictions matched the multimodal baseline.

The ablation results (Table 4) provide empirical evidence for our hypothesis of visual dominance in low-resource settings. In Subtask A (Hate Speech Detection), SigLIP exhibited a strong image bias (34 instances), significantly outpacing the minimal text bias (6 instances). This suggests that the model relies extensively on visual signals for hate detection, effectively bypassing the noisy or code-mixed Nepali textual stream.

Conversely, in Subtask B (Sentiment Classification), the modality reliance was considerably more balanced. The ablation identified 12 image-biased and 17 text-biased cases.However, the vast majority (104 instances) represented true multimodal synergy.

7. Conclusion

This paper has compared different unimodal, early-fusion, and foundation models of the CHiPSAL 2026 Shared Task on Nepali meme moderation. Contrary to the intuitive expectation that explicit text-vision fusion is necessary, we find that single visual architectures and pre-aligned foundation models carry the major semantic burden in this low-resource, high-noise environment. In particular, the Vision Transformer (ViT) had the largest F1-score (0.6278) on binary Hate Speech Detection

with exploiting the visual semiotics, and the SigLIP foundation model excelled at the multi-class Sentiment Analysis subtask (0.5481 F1-score) with the strong cross-modal synergy. These findings finally show that the performance of noisy OCR-extracted text with powerful visual features through standard late fusion is actually worse, which emphasizes the importance of noise-resilient text encoders or more intensive utilization of vision-language models with inherent alignment on low-resource languages.

8. Limitations

Despite establishing encouraging baselines, our study presents several limitations. The most prominent bottleneck stems from OCR transcription errors and the casual, code-mixed nature of Nepali internet slang, which significantly degraded the quality of the textual embeddings utilized by our early-fusion models. We did not employ explicit text-denoising or image-restoration techniques prior to text extraction. This decision was twofold: first, there is a current lack of robust, off-the-shelf denoising or spell-correction pipelines optimized for code-mixed, low-resource Nepali internet data. Second, within meme culture, visual noise (e.g., low resolution, grain, compression artifacts, or "deep-fried" effects) is frequently considered part of the intentional aesthetic. Aggressively removing these artifacts to improve OCR accuracy risks destroying subtle semiotic cues that heavily influence the meme's overall sentiment or hateful intent.

Furthermore, our interpretability analysis indicates that visual models, such as ViT, are over-reliant on specific visual markers like emojis. Consequently, these models often fail to identify malicious intent in highly contextual memes where the subtle synergy between text and image is the primary driver of the sentiment.

Finally, our experiments were constrained by a relatively small training dataset. This data scarcity likely contributed directly to the underperformance of our explicit early-fusion architectures compared to unimodal baselines and pre-aligned foundation models, the fusion networks simply lacked sufficient examples to effectively learn complex cross-modal alignments from scratch. This constrained our models' capacity to detect nuanced inter-modal semantic dissonance and generalize across the rapidly evolving landscape of Nepali digital culture. To bypass the fragile text-extraction phase entirely and mitigate OCR-induced noise without destroying the underlying visual aesthetic, future research should prioritize the use of true OCR-free Vision-Language Foundation Models (such as Donut or PaliGemma). These specialized, end-to-end architectures represent the most promising primary next step for multimodal comprehension in low-resource, high-noise environments.

9. References

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443.

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and flexible image augmentations. *Information*, 11(2).

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John P. McCrae. 2020. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John Philip McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, Kyiv. Association for Computational Linguistics.

Alexis Conneau et al. 2020. Unsupervised cross-lingual representation learning at scale. *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

Alexey Dosovitskiy et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

Kaiming He et al. 2016. Deep residual learning for image recognition. *CVPR*.

Simran Khanuja et al. 2021. Muril: Multilingual representations for indian languages. *EMNLP*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020a. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624. Curran Associates, Inc.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020b. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624.

Zhuang Liu et al. 2022. A convnet for the 2020s. *CVPR*.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Chiranjibi Sitaula, Anish Basnet, Ashish Mainali, and Tej Bahadur Shahi. 2021. Deep learning-based methods for sentiment analysis on nepali covid-19-related tweets. *Computational Intelligence and Neuroscience*, 2021(1):2158184.

Surendrabikram Thapa, Kritesh Rauniyar, Shuvam Shiwakoti, Sweta Poudel, Usman Naseem, and Mehwish Nasim. 2023. Nehate: Large-scale annotated data shedding light on hate speech in nepali local election discourse. *Frontiers in Artificial Intelligence and Applications*, 372:2346–2353.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Xiaohua Zhai et al. 2023. Sigmoid loss for language image pre-training. *ICCV*.

Appendix A. Detailed Experimental Setup

Model Type	Learning Rate / LoRA
Fusion	1×10^{-5} or LoRA ($r = 16$, $\alpha = 32$)
Image-Only	1×10^{-5} or LoRA ($r = 16$, $\alpha = 32$)
Text-Only	LoRA Fine-tuning ($r = 16$, $\alpha = 32$)
Classifier	1×10^{-4} (All models)

Table 5: Model-specific learning rates and LoRA parameters.

linus@CHiPSAL 2026: Multimodal Hate Speech and Sentiment Detection in Low-Resource Memes using Late-Fusion Hybrid Architecture

Sunil Regmi[1], **Bipesh Subedi**[2], **Saugat Singh**[2], **Suman Shrestha**[2]
[1] Department of Artificial Intelligence, Kathmandu University
[2] Department of Computer Science and Engineering, Kathmandu University
sunilregmi233@gmail.com, bipeshrajsubedi@gmail.com, saugat.singh09@gmail.com, shrestha.suman@ku.edu.np

Abstract

The increased sharing of memes on social media creates serious challenges for automated moderation, especially in low-resource and code-mixed languages such as Nepali. In this paper, we present our system for the CHiPSAL 2026 Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes. We propose a late-fusion hybrid architecture that combines OpenAI's Vision Transformer (CLIP ViT-B/32) with a domain-specific Nepali language model (NepBERTa) to capture both visual features and linguistic information. To address data scarcity, we introduce a cross-task label mapping and data augmentation strategy between the hate speech and sentiment datasets. By applying controlled hyperparameter settings and balanced loss optimization, our framework achieved a Macro F1 score of 0.8052 on Subtask A (Hate Speech Detection) and 0.6881 on Subtask B (Sentiment Analysis) in the official CodaBench evaluation, demonstrating the effectiveness of the proposed multimodal approach. The system was developed as part of a shared task under strict evaluation constraints where test labels were not publicly available.

Keywords: Multimodal, Hate Speech Detection, Sentiment Analysis, Code-Mixed Nepali, Late-Fusion

1. Introduction

Memes have become a distinctive and powerful form of digital communication, blending visual and textual information to convey a vast spectrum of emotions, opinions, and political critiques(Elahi et al., 2023a). As these digital constructs are deeply ingrained in internet culture, they serve as essential tools for sharing opinions and constructing collective identities across cultural boundaries(Konyspay et al., 2025). However, the rapid evolution of social media has also facilitated the proliferation of harmful content, such as hate speech, offensive trolling, and misinformation, which poses significant challenges for automated moderation systems(Mishra et al., 2023).

Understanding the core sentiment and intention behind memes is a complex task due to their frequent reliance on nuanced aggression, satire, and the complex relationship between words and visuals. Although significant studies have been carried out on memes in the English language, languages with fewer resources, such as Nepali, Hindi, Bengali and Tamil, have been investigated much less. These languages encounter specific challenges, such as a significant shortage of benchmark datasets, the absence of standardized optical character recognition (OCR) tools, and the common practice of using code-mixed or code-switched content (like "Nenglish," "Hinglish," "Banglish," or "Tanglish"), where native dialects are represented using English characters. To tackle these challenges effectively, research has progressively moved towards multimodal methods. Findings indicate that unimodal models, which examine text or images on their own, generally fall short as they overlook the contextual signals that arise from the interplay of both modalities.

Recent research shows that multimodal fusion, the collaborative combination of visual and textual elements greatly enhances the precision of sentiment analysis and trolling identification(Kannan and Rajalakshmi, 2022). Through the use of sophisticated deep learning models and Explainable AI (XAI) methods, researchers aim to create more resilient and understandable frameworks that elucidate the intricate social and emotional interactions of memes in linguistic contexts that lack resources(Elahi et al., 2023a).Recognizing these challenges in low-resource South Asian languages, the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026) introduced a shared task focused on detecting hate speech and analyzing sentiment in Nepali-embedded images (Thapa et al., 2026; Sarveswaran et al., 2026). The task provides a structured benchmark for evaluating multimodal systems in code-mixed Nepali contexts, encouraging the development of robust approaches for harmful content detection in resource-constrained settings.

This paper outlines our team's methodological approach to hate speech detection and sentiment analysis on Nepali memes. Given the complexity

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 267–274
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

of standard computational social science analysis paradigms (Parihar et al., 2021; Thapa et al., 2025), we developed a structurally augmented Late-Fusion framework. By utilizing **NepBERTa** for high-fidelity textual extraction alongside **CLIP** for robust visual anchoring, we significantly bridge the localized context gap often encountered when analyzing text-embedded Nepali images. We also introduce a methodology for cross-task augmentation that maximizes the utility of sparse annotated data, proving that mapping implicit labels across overlapping domains provides significant stabilization during training.

2. Related Work

Multimodal hate speech and sentiment analysis in memes has evolved significantly over the past decade. Early approaches primarily focused on unimodal textual analysis using traditional machine learning algorithms such as Support Vector Machines (SVMs) and Naive Bayes classifiers, later transitioning to recurrent neural networks (RNNs) and standard Transformer architectures. While these methods achieved reasonable performance in text-only hate detection, they struggle in multimodal settings such as memes, where textual content may appear benign in isolation but becomes offensive when interpreted alongside visual context.

To address this limitation, subsequent research introduced multimodal fusion frameworks that combine textual and visual features. For instance, (Dubey et al., 2025) curated a dataset of 9,262 Hindi memes and employed OCR-based text extraction followed by multilingual embeddings (LASER, LaBSE) and CNN-based image encoders (e.g., VGG16, ResNet18), achieving 80% accuracy through late fusion. Similarly, (Karim et al., 2022) extended the Bengali Hate Speech Dataset with 4,500 labeled memes and demonstrated that integrating CNN backbones with transformer-based language models (RoBERTa, XLM-R, BanglaBERT) improves performance, achieving an F1-score of 0.79 and MCC of 0.808. (Rajput et al., 2022) further explored code-switched political memes using a CNN–LSTM multimodal framework, reporting an F1-score of 0.792 and highlighting the challenges of multilingual and politically nuanced content.

More recent advancements reflect a paradigm shift toward joint multimodal representation learning through large-scale vision–language pretrained models. Architectures such as VisualBERT, LXMERT, and CLIP align textual and visual embeddings in a shared latent space, enabling cross-modal reasoning without extensive task-specific feature engineering. A CLIP-based architecture with prompt engineering achieved 87.42% accuracy and 90.13% F1-score in a zero-shot hate speech detection setting (Arya et al., 2024). In the context of multimodal hate speech event detection, (El-Sayed and Nasr, 2024) combined CLIP with transformer-based text models such as HateBERT, RoBERTa, and XLM-R on the CrisisHateMM dataset, achieving an F1-score of 0.7703.

Parallel efforts in multimodal sentiment analysis have shown similar trends. (Elahi et al., 2023b) introduced the MemoSEN dataset of 4,372 Bengali memes categorized as positive, negative, or neutral, using ResNet50 and BanglishBERT to achieve a weighted F1-score of 0.71. Similarly, (Guleria et al., 2024) analyzed English and Hinglish memes using a combination of RoBERTa and CLIP, reporting an accuracy of 0.82.

Despite these advances, multimodal hate speech detection in South Asian languages remains relatively underexplored, particularly for low-resource languages such as Nepali. Existing studies emphasize the importance of localized language models for handling code-switched, morphologically rich, or transliterated text, as generalized multilingual models may inadequately capture linguistic nuances. Motivated by these gaps, we adopt a hybrid paradigm that integrates the generalization strength of vision–language pretraining with the linguistic specificity of a regional transformer. Specifically, we fuse the visual backbone of CLIP with NepBERTa, a RoBERTa-based model tailored for Nepali language understanding, enabling robust multimodal representation learning in low-resource meme datasets.

3. Dataset & Task

The primary data used is the **CHiPSAL 2026 Dataset**, which compiles low-resource, code-mixed Nepali memes. The dataset draws from foundational multimodal datasets and task architectures in the domain (Thapa et al., 2025a,b; Bhandari et al., 2023). Due to the linguistic diversity of South Asia, the dataset inherently contains heavy code-switching - primarily a mixture of Devanagari script, phonetically transliterated English written in Devanagari, and English terms written natively. This structural complexity introduces substantial challenges for standard parsers. The shared task is divided into:

- **Subtask A (Hate Speech Detection):** A binary classification task to identify whether a meme contains "Hate" (label 1) or "No-Hate" (label 0). Hate speech in this context constitutes explicit slurs, targeted attacks, violence incitement, or profoundly offensive rhetoric targeting marginalized groups based on race, gender, religion, or caste. Determining this

often requires inferring the intent behind the overlaying text when combined with an inciting conceptual image.

- **Subtask B (Sentiment Analysis):** A three-way classification task categorizing memes into "Negative" (label 0), "Neutral" (label 1), or "Positive" (label 2). This task maps the generalized emotional polarity of the meme. Note that positive sentiment imagery overlaid with highly sarcastic text frequently maps to a "Negative" ground truth, demanding deep multi-task analytical capabilities.

3.1. Cross-Task Data Creation and Augmentation

To address data scarcity and improve training robustness, we explore a cross-task data augmentation strategy that leverages the overlap between sentiment polarity and hate speech signals in meme data. However, this relationship is not always straightforward. In meme contexts, particularly in low-resource and code-mixed settings, sentiment labels can be ambiguous. For example, samples labeled as "Positive" may sometimes express sarcasm or irony, which can align with underlying hateful intent depending on context. Therefore, the connection between sentiment and hate labels should be treated as approximate and context-dependent rather than strictly equivalent. We highlight this as a potential direction that requires more careful investigation in future work.

In this work, we adopt a simple heuristic-based cross-task label mapping to construct augmented datasets. Specifically, we approximate the alignment by treating negative sentiment as indicative of hate and positive sentiment as indicative of non-hate. While this simplification does not capture all nuances, it provides a practical way to expand the training data. The detailed statistics for both subtasks, including original, augmented, and validation splits, are reported in Tables 1 and 2.

- **Augmenting Subtask A:** We extend the Subtask A dataset using samples from Subtask B. Since Subtask A is a binary classification task, we first remove all “Neutral” (1) instances from Subtask B. The remaining samples are mapped as follows: Subtask B “Negative” (0) $\rightarrow$ Subtask A “Hate” (1), and Subtask B “Positive” (2) $\rightarrow$ Subtask A “No-Hate” (0).
- **Augmenting Subtask B:** Similarly, we augment Subtask B using samples from Subtask A by projecting binary labels onto the sentiment space. Specifically, Subtask A “No-Hate” (0) is mapped to Subtask B “Positive” (2), while Subtask A “Hate” (1) is mapped to Subtask B “Negative” (0).

Split	Total	No Hate	Hate
train	1068	348	720
train(augmented)	1656	595	1061
validation	133	35	98

Table 1: Subtask A (Hate Speech Detection) Dataset Statistics

Split	Total	-ve	Neutral	+ve
train	1061	341	473	247
train(aug.)	2129	1061	473	595
validation	133	39	65	29

Table 2: Subtask B (Sentiment Analysis) Dataset Statistics

4. Methodology

4.1. Data Preprocessing

Before inputting the data streams into the multimodal architecture, rigorous pre-processing pipelines were established for both textual and visual modalities.

Images were first normalized according to CLIP prerequisites, which enforce specific channel-wise mean and standard deviation metrics to align the visual geometry to the original pre-training subspace. To establish extreme robustness against standard formatting variations seen in low-resolution internet memes, dynamic PyTorch visual augmentations were applied algorithmically bounding the training datasets. We used a combination of `RandomResizedCrop` (to scale against shifting aspect ratios), `RandomHorizontalFlip`, `RandomRotation` up to 15° (to counter angled text constraints), and aggressive `ColorJitter` (dynamically adjusting brightness, contrast, and saturation variations common in recaptured media).

Text elements native to the imagery were tokenized using the `Rajan/nepbertaTorch`[1] Subword Tokenizer. Standard generic preprocessing approaches often employ basic whitespace delimitation which catastrophically breaks morphological Nepali structures. Our tokenizer preserves localized contextual information, encoding sub-words efficiently. Truncation was enforced at a static sequence length limit of 128 tokens, zero-padding the subsequent sequence fields. This uniform bounding was essential for scaling parallel processing during the highly constrained hyperparameter tuning phase.

4.2. Model Architecture

To ensure accurate localization against native Nepali dialects while maintaining robust visual con-

[1] https://huggingface.co/Rajan/nepbertaTorch

text, our core modeling strategy leverages a **Hybrid Late-Fusion** architecture matching a specialized text extraction module with a broad-domain visual foundation model.

4.2.1. Visual Feature Extraction

We utilize OpenAI's CLIP (ViT-B/32) (Radford et al., 2021) to extract high-quality visual representations. CLIP's Vision Transformer parses the input image into a sequence of non-overlapping patches, which are linearly embedded and prepended with a learnable classification (`[CLS]`) token. The final hidden state corresponding to this `[CLS]` token serves as the global visual representation, denoted as $v \in \mathbb{R}^{d_v}$, where $d_v = 512$. By leveraging CLIP, the model inherently possesses strong zero-shot capabilities and robust structural understanding of diverse, internet-native imagery, which is crucial for meme analysis.

4.2.2. Textual Feature Extraction

For the textual modality, we employ **NepBERTa**, a transformer model specifically pre-trained on a vast corpus of Nepali text. Traditional multilingual models like mBERT or Multilingual-CLIP often use byte-pair encodings that poorly align with Devanagari script, resulting in fragmented subwords and diluted semantic representation (Thapa et al., 2025). NepBERTa, conversely, captures the morphological intricacies of Nepali. The tokenized input sequence is passed through the transformer layers, and the embedding corresponding to the sequence-level `[CLS]` token is extracted as the text representation $t \in \mathbb{R}^{d_t}$, where $d_t = 768$.

4.2.3. Late-Fusion and Classification

The core of our approach lies in the late-fusion mechanism. Premature fusion can lead to one modality dominating the gradient updates, especially when textual cues (like explicitly offensive words) provide stronger immediate signals than complex visual context. To prevent this, the extracted features are independently L2-normalized to ensure stable gradient scales:

$$\hat{v} = \frac{v}{\|v\|_2}, \quad \hat{t} = \frac{t}{\|t\|_2} \tag{1}$$

These normalized embeddings are then concatenated into a dense 1280-dimensional joint representation z:

$$z = \hat{v} \oplus \hat{t} \in \mathbb{R}^{1280} \tag{2}$$

To synthesize and stabilize the multimodal gradients before classification, a `LayerNorm` operation is applied to z. The fused state then passes through a dense multi-layer perceptron (MLP) structured as follows:

1. Linear transformation reducing dimensionality from 1280 to 512, followed by a GELU activation and Dropout (0.3).
2. Linear transformation from 512 to 128, followed by a GELU activation and Dropout (0.2).
3. Final Linear projection from 128 to the target num_labels, producing the output logits $\hat{y}$.

4.3. Training Setup

Training optimizations strictly leveraged adaptive `AdamW` implementations. Realizing the danger of catastrophic forgetting in pre-trained instances, we deployed significantly lower learning rates for the backbones (2×10^{-6} to 3×10^{-6}) while implementing a steeper gradient curve for the classifier head (1×10^{-4} to 2×10^{-4}), managed via a `OneCycleLR` scheduler.

Furthermore, we instituted aggressive explicit class weights (e.g., `[1.0, 4.0, 1.0]`) to handle "Neutral" label sparsity in Subtask B. Additionally, `label_smoothing=0.1` was integrated to throttle overconfidence biases favoring major label instances. Hardware parameters utilized active Gradient Accumulation to guarantee high generalized batch-resolutions.

4.4. Hyperparameter Optimization and Setup

To discover the optimal learning configuration that balances the pre-trained weights without causing catastrophic forgetting, we conducted structured grid searches. Our optimization tracked parameter subsets varying the backbone learning rates from 1×10^{-6} to 5×10^{-6} and head learning rates from 5×10^{-5} up to 2×10^{-4}. We implemented a mini-batch processing logic mapping target batch sizes to dynamic `gradient_accumulation_steps`, ensuring consistent optimization geometries extending beyond standard VRAM limitations. The final models were evaluated after 50 epochs utilizing an `AdamW` optimizer scaled harmoniously via the `OneCycleLR` scheduler.

5. Results & Discussion

Task	Acc.	F1 Mac.	Prec.	Recall
Sub. A	0.8209	**0.8052**	0.7974	0.8202
Sub. B	0.6842	**0.6881**	0.6979	0.7013

Table 3: Performance on Official Test Benchmarks

The models were evaluated using the official CodaBench evaluation system, where performance

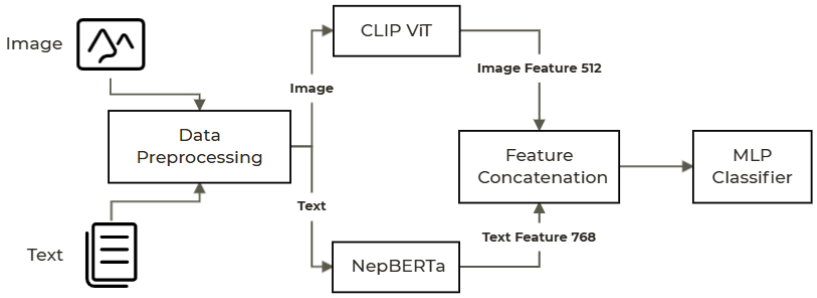


Figure 1: Overall architecture of the proposed multimodal framework combining CLIP-based visual features with NepBERTa textual embeddings using late fusion and regularized classification layers.

was measured in terms of Accuracy, Macro-F1, Precision, and Recall on the respective test sets. Detailed results are reported in Table 3.

Integrating cross-task augmented data into Subtask A allowed us to explicitly amplify the boundaries between Hate and No-Hate classes. By completely removing Neutral instances from the augmentation process, we avoided introducing ambiguous signals, which contributed to Subtask A achieving a peak Test F1 of 0.8052. This confirms the effectiveness of a constrained cross-task augmentation strategy that leverages overlapping domains while controlling for label noise.

In contrast, Subtask B remains more challenging due to its three-class structure. While Subtask A benefits from explicit binary boundaries, Subtask B requires distinguishing among Positive, Negative, and Neutral classes. Even though Neutral instances were not added via augmentation, the presence of this intermediate class inherently introduces ambiguity, as subtle expressions and code-mixed nuances often overlap with both Positive and Negative labels. This likely contributes to the relatively lower Macro F1 score of 0.6881 compared to Subtask A.

To mitigate overfitting and improve generalization, techniques such as `label_smoothing` and dropout were applied. These methods were particularly helpful for Subtask B, where noisy textual signals from memes can otherwise cause the model to memorize the training data. Interestingly, the Subtask B model achieved a Macro F1 score on the official test set that was higher than its internal validation score (0.6881 vs. 0.6607), demonstrating improved generalization to unseen examples.

During hyperparameter tuning, we observed that setting the learning rate too high for the pre-trained backbones led to drastic performance drops. When backbone learning rates exceeded 2×10^{-6}, F1 scores fell to around 0.55, likely because the pre-trained representations in NepBERTa and CLIP were disrupted. The best performance was achieved with a backbone learning rate of 3×10^{-6} and a higher learning rate of 2×10^{-4} for the classification head. Keeping backbone learning rates low allowed the classification head to effectively learn feature fusion without destroying pre-trained knowledge.

While the model achieves competitive performance, there remains room for improvement, particularly in handling complex multimodal interactions where the relationship between text and visual content is complex or context-dependent. These challenges are inherent to the task and highlight potential directions for future work.

5.1. Ablation Study

To better understand the contribution of each component, we perform ablation experiments on the validation split as shown in Table 4. Due to shared task constraints, ground-truth labels for the official test set are not publicly available. Therefore, all comparative experiments are conducted using validation data, which was also used for model selection prior to final submission.

Model	Subtask A (F1)	Subtask B (F1)
CLIP-only	0.54	0.59
NepBERTa-only	0.63	0.60
Fusion (Ours)	0.78	0.66

Table 4: Ablation study on validation set

The results show that the fusion model outperforms both unimodal variants across both subtasks. CLIP-only achieves the lowest scores, indicating that visual features alone are insufficient for accurate meme understanding. NepBERTa-only performs better, suggesting textual information plays a stronger role. However, the fusion model achieves the highest performance, improving Subtask A from 0.63 to 0.78 and Subtask B from 0.60 to 0.66.

This confirms that combining visual and textual features helps capture additional context unavailable to single-modality models. The observation aligns with prior multimodal learning research,

where integrated vision–language representations improve tasks requiring joint reasoning over text and images (Dubey et al., 2025; Karim et al., 2022). Recent aligned vision-language models further support the efficacy of multimodal fusion, demonstrating strong generalization across diverse tasks.

Overall, the results highlight the importance of multimodal fusion and controlled cross-task augmentation for complex meme datasets, especially in low-resource and code-mixed settings. The final fusion model was selected based on validation performance and submitted to the official evaluation server, where it achieved the reported test scores.

6. Conclusion

This work demonstrates that combining a domain-specific language model (NepBERTa) with a strong visual encoder (CLIP) provides an effective baseline for multimodal understanding in low-resource, code-mixed meme data. The proposed late-fusion approach achieved competitive performance in the CHiPSAL 2026 shared task. Our findings suggest that cross-task augmentation can provide additional training signals in low-data settings, although its effectiveness depends on the quality of label mapping. The results also highlight the importance of textual features, while visual features provide complementary context.

Future work can explore improved OCR systems, better handling of sarcasm, and more advanced multimodal fusion techniques.

Data and Model: This work utilizes the CHiPSAL 2026 shared task dataset, publicly available at https://github.com/therealthapa/chipsal26-memes . The pretrained model used in this study are available in hugging face. The implementation code and the models will be made publicly available in a repository (https://github.com/sunilRegmi-ai/chipsal2026) following publication .

7. Limitations

A key limitation of our work is that ablation studies could not be performed on the official test set, as ground-truth labels were not publicly available during the competition. Although we report unimodal and fusion comparisons on the validation set, the exact contribution of each component on the test set remains uncertain. The proposed cross-task label mapping is heuristic and assumes a direct relationship between sentiment and hate speech (e.g., positive sentiment mapped to non-hate and negative sentiment mapped to hate). While this assumption is intuitive, it may not always hold in practice, particularly in cases such as sarcasm, implicit hate, or context-dependent expressions, which may introduce noisy or ambiguous training signals.While we provide ablation results comparing unimodal and fusion models, we did not isolate the effect of cross-task augmentation separately. Therefore, its exact contribution to performance improvement cannot be fully quantified. We also did not experimentally evaluate alternative fusion strategies such as early fusion or cross-attention. Our choice of late fusion is based on simplicity and stability rather than direct empirical comparison with other fusion methods. In addition, the system depends on OCR-extracted text, making it sensitive to OCR errors and degraded image quality. When text extraction fails or produces incorrect tokens, model performance can degrade significantly.

Finally, the model struggles with sarcasm and culturally nuanced expressions, which are common in meme data. Addressing these challenges would require more advanced multimodal reasoning and improved language understanding in low-resource settings.

8. Ethics Statement

This work uses the CHiPSAL 2026 shared task dataset, which is publicly available for research. Meme-based data may contain sensitive or biased content, and models trained on such data can reflect or amplify these biases. The model also relies on pretrained representations from the Hugging Face ecosystem, which may introduce additional biases.

In terms of limitations, the model may not generalize well to unseen or culturally diverse memes and does not explicitly address bias mitigation, robustness, or interpretability.

9. Bibliographical References

Greeshma Arya, Mohammad Kamrul Hasan, Ashish Bagwari, Nurhizam Safie, Shayla Islam, Fatima Rayan Awad Ahmed, Aaishani De, Muhammad Attique Khan, and Taher M. Ghazal. 2024. Multimodal hate speech detection in memes using contrastive language-image pre-training. *IEEE Access*, 12:22359–22375.

Kriti Dubey, Vaishnavi Srivastava, Garima Sharma, Nonita Sharma, Deepak Sharma, Uttam Ghosh, Osama Alfarraj, and Amr Tolba. 2025. Multimodal detection of offensive content in hindi memes. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Ahmed El-Sayed and Omar Nasr. 2024. Aast-nlp at multimodal hate speech event detection 2024

: A multimodal approach for classification of text-embedded images based on clip and bert-based models. In *CASE*.

Kazi Toufique Elahi, Tasnuva Binte Rahman, Shakil Shahriar, Samir Sarker, Sajib Kumar Saha Joy, and Faisal Muhammad Shah. 2023a. Explainable multimodal sentiment analysis on bengali memes.

Kazi Toufique Elahi, Tasnuva Binte Rahman, Shakil Shahriar, Samir Sarker, Sajib Kumar Saha Joy, and Faisal Muhammad Shah. 2023b. Explainable multimodal sentiment analysis on bengali memes. In *Proceedings of the International Conference on Computer and Information Technology (ICCIT)*.

Aishvi Guleria, Kamya Varshney, Garima Pahwa, Shreya Singhal, and Nonita Sharma. 2024. Multimodal sentiment analysis of english and hinglish memes. *Multimedia Tools and Applications*.

Ramesh Kannan and Ratnavel Rajalakshmi. 2022. Multimodal code-mixed Tamil troll meme classification using feature fusion. In *Proceedings of the First Workshop on Multimodal Machine Learning in Low-resource Languages*, pages 1–8, IIIT Delhi, New Delhi, India. Association for Computational Linguistics.

Md. Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Md. Shajalal, and Bharathi Raja Chakravarthi. 2022. Multimodal hate speech detection from bengali memes and texts. In *Proceedings of the International Conference on Speech and Language Technologies for Low-Resource Languages*.

Aidos Konyspay, Pakizar Shamoi, Malika Ziyada, and Zhusup Smambayev. 2025. Meme similarity and emotion detection using multimodal analysis. In *2025 International Conference on Activity and Behavior Computing (ABC)*, page 1–10. IEEE.

Shreyash Mishra, S Suryavardan, Parth Patwa, Megha Chakraborty, Anku Rani, Aishwarya Reganti, Aman Chadha, Amitava Das, Amit Sheth, Manoj Chinnakotla, Asif Ekbal, and Srijan Kumar. 2023. Memotion 3: Dataset on sentiment and emotion analysis of codemixed hindi-english memes.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Kshitij Rajput, Raghav Kapoor, Kaushal Rai, and Preeti Kaur. 2022. Hate me not: Detecting hate inducing memes in code switched languages. *arXiv preprint arXiv:2204.11356*.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Surabhi Adhikari, Hariram Veeramani, Mehwish Nasim, and Usman Naseem. 2025. Large language models (llm) in computational social science: prospects, current state, and challenges. *Social Network Analysis and Mining*, 15(1):1–30.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

10. Language Resource References

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

ZeroR@CHiPSAL 2026: Two-Stage Vision-Language Adaptation with Contrastive Learning for Nepali Meme Classification

Nitiz Khanal
Pulchowk Campus, Institute of Engineering, Tribhuvan University
Lalitpur, Nepal
khanalnitij20@gmail.com

Abstract

This paper presents our system for the CHiPSAL 2026 shared task on multimodal hate speech and sentiment detection in Nepali memes. We address both subtasks: binary hate speech classification and three-class sentiment analysis. Our approach adapts the Robust Adaptation of Hateful Meme Detection (RA-HMD) framework using Qwen3-VL-8B-Instruct, a state-of-the-art vision-language model with native Devanagari support. We employ a two-stage training pipeline: (1) LoRA fine-tuning with an MLP projection head for generative classification, and (2) contrastive backbone fine-tuning with supervised InfoNCE loss. We handle class imbalance through minority oversampling, image augmentation, and focal loss. At inference, we ensemble Stage 1 token probabilities with Stage 2 classifier scores using validation-tuned weights. Our end-to-end approach eliminates error propagation from separate OCR and translation pipelines by leveraging the model's native Devanagari understanding. Our system achieved **2nd place** on hate speech detection (F1: 0.797) and **4th place** on sentiment analysis (F1: 0.518). We provide detailed ablations, error analysis, and insights into adapting large vision-language models for low-resource South Asian languages.

Keywords: hate speech detection, sentiment analysis, vision-language models, Nepali, multimodal, contrastive learning, low-resource languages

1. Introduction

The proliferation of memes on social media platforms has created new challenges for content moderation systems. Unlike traditional text or image classification, memes require understanding the complex interplay between visual and textual elements, where meaning often emerges from their combination rather than either modality alone (Kiela et al., 2020). A seemingly innocuous image paired with specific text can convey hate, while aggressive text on a humorous image might be entirely benign. This multimodal reasoning challenge is further compounded for low-resource languages, where annotated datasets are scarce and pre-trained models have limited exposure to the language and its cultural context (Parihar et al., 2021).

Nepali, spoken by approximately 32 million people, presents unique challenges for automated content moderation. The language uses Devanagari script, which is embedded directly into meme images rather than appearing as separate text. Cultural references, humor styles, and implicit meanings rooted in Nepali society add layers of complexity that general-purpose models may not capture. Furthermore, the code-mixing of Nepali with English, common in social media, creates additional processing challenges.

The Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2026), co-located with LREC-COLING 2026, addresses these challenges through shared tasks focused on South Asian language processing (Sarveswaran et al., 2026). We participated in the multimodal hate and sentiment detection task (Thapa et al., 2026), which provides carefully curated Nepali meme datasets collected from social media platforms (Thapa et al., 2025b,c; Bhandari et al., 2023). The shared task comprises two subtasks:

- **Subtask A – Hate Speech Detection**: Binary classification of memes as hateful (content targeting groups based on ethnicity, religion, gender, or caste, or promoting violence/discrimination) or safe.
- **Subtask B – Sentiment Analysis**: Three-class classification of memes into negative, neutral, or positive sentiment categories.

Our approach leverages recent advances in vision-language models (VLMs), specifically adapting the Robust Adaptation of Large Multimodal Models (RA-HMD) framework (Mei et al., 2025) for low-resource Nepali content. Rather than constructing complex pipelines involving OCR extraction, machine translation, and separate text/image encoders, we leverage Qwen3-VL-8B-Instruct (Wang et al., 2024), a state-of-the-art VLM with native support for multiple scripts including Devanagari. This end-to-end approach eliminates error propagation that would occur in pipeline-based systems.

Our key contributions are:

1. A two-stage training framework combining generative fine-tuning with contrastive learning, adapted from RA-HMD for low-resource multimodal classification.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 275–283
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

2. Task-specific configurations addressing the distinct challenges of binary hate speech detection versus three-class sentiment analysis.

Our system achieved **2nd place** on hate speech detection (F1: 0.797) and **4th place** on sentiment analysis (F1: 0.518), demonstrating the effectiveness of adapting modern VLMs for low-resource multimodal tasks.

2. Related Work

2.1. Multimodal Hate Speech Detection

The challenge of detecting hate speech in multimodal content gained significant attention with the Facebook Hateful Memes Challenge (Kiela et al., 2020), which demonstrated that simple fusion of unimodal representations is insufficient. The challenge showed that even state-of-the-art vision and language models struggled when hate was conveyed through subtle interactions between image and text.

Subsequent work has explored various fusion strategies, from early fusion approaches that concatenate features to more sophisticated cross-modal attention mechanisms. Pramanick et al. (2021) introduced methods for detecting harmful memes and identifying their targets, while Bhandari et al. (2023) extended multimodal hate speech analysis to crisis contexts with the CrisisHateMM dataset, providing annotation schemas for directed and undirected hate.

More recently, Mei et al. (2024) proposed retrieval-guided contrastive learning for hateful meme detection, using similar training samples to guide representation learning. This work was extended by Mei et al. (2025) with the RA-HMD framework, which combines LoRA fine-tuning with rationale-aware contrastive learning, achieving state-of-the-art results on multiple English hateful meme datasets including HatefulMemes, MAMI, and PrideMM. Our work adapts this framework for low-resource Nepali content.

2.2. Vision-Language Models

The emergence of large vision-language models has transformed multimodal understanding. Early approaches like CLIP (Radford et al., 2021) learned joint image-text representations through contrastive learning on web-scale data. More recent instruction-tuned models like LLaVA (Liu et al., 2024) and the Qwen-VL family (Wang et al., 2024) can follow complex multimodal instructions and generate detailed responses.

Qwen3-VL-8B-Instruct, which we use as our backbone, represents a significant advancement in multimodal understanding. The model supports dynamic image resolution, multiple languages including those using non-Latin scripts, and demonstrates strong performance across diverse vision-language benchmarks. Critically for our application, it has native support for Devanagari script, enabling end-to-end processing of Nepali text embedded in images without requiring separate OCR.

2.3. Parameter-Efficient Fine-Tuning

Fine-tuning large models on downstream tasks is computationally expensive and can lead to catastrophic forgetting. Low-Rank Adaptation (LoRA) (Hu et al., 2022) addresses this by freezing pre-trained weights and injecting trainable rank decomposition matrices into transformer layers. This approach reduces trainable parameters by orders of magnitude while maintaining competitive performance.

For vision-language models, LoRA has proven particularly effective, allowing adaptation to new tasks and domains without the computational cost of full fine-tuning. We apply LoRA to all linear projections in Qwen3-VL, enabling efficient adaptation to Nepali meme classification.

2.4. Contrastive Learning for Classification

Supervised contrastive learning (Khosla et al., 2020) extends the self-supervised contrastive paradigm by leveraging label information. The objective pulls together representations of samples from the same class while pushing apart representations from different classes. The InfoNCE loss (van den Oord et al., 2018) provides a principled framework for this objective.

For content moderation tasks, contrastive learning helps learn discriminative representations that capture subtle differences between classes. Mei et al. (2024) and Mei et al. (2025) demonstrated its effectiveness for hateful meme detection, and we adapt this approach for Nepali content.

2.5. Low-Resource Language Processing

Processing low-resource languages presents unique challenges due to limited training data, fewer pre-trained resources, and distinct linguistic characteristics. Large language models have shown promise for such languages through cross-lingual transfer (Thapa et al., 2025a), but multimodal tasks in low-resource settings remain underexplored.

The CHiPSAL workshop series (Sarveswaran et al., 2026) aims to address these gaps for South Asian languages. Prior work on Nepali NLP has focused primarily on text classification and named entity recognition, with multimodal analysis receiv-

ing limited attention. Our work contributes to this emerging area by demonstrating effective adaptation of state-of-the-art VLMs for Nepali multimodal content.

3. Dataset and Task Description

3.1. Dataset Overview

The shared task provides Nepali meme datasets collected from various social media platforms (Thapa et al., 2025b,c). The memes contain Devanagari text embedded in images, often with code-mixing between Nepali and English. The annotation follows established protocols for multimodal hate speech (Bhandari et al., 2023).

3.2. Subtask A: Hate Speech Detection

The hate speech dataset contains approximately 1,068 memes with binary labels:

- **Hate (1)**: Content that attacks, demeans, or incites hatred against individuals or groups based on protected characteristics including ethnicity, religion, gender, caste, nationality, or disability. This also includes content promoting violence or discrimination.
- **Safe (0)**: Content that does not contain hate speech, including neutral information, positive messages, and humor that does not target protected groups.

The dataset exhibits class imbalance, with safe memes being more frequent than hateful ones. This imbalance reflects the natural distribution of content on social media but poses challenges for classification.

3.3. Subtask B: Sentiment Analysis

The sentiment dataset contains memes labeled with three categories:

- **Negative (0)**: Content expressing criticism, mockery, sadness, anger, frustration, disappointment, or pessimism.
- **Neutral (1)**: Factual information, observations, or content without clear emotional valence.
- **Positive (2)**: Content expressing joy, humor, encouragement, celebration, love, hope, or optimism.

The neutral class dominates the distribution, creating a significant class imbalance challenge. Additionally, the boundary between categories can be subjective, particularly for humorous content that may express negativity in a playful manner.

3.4. Evaluation Metric

Both subtasks use macro F1-score as the primary evaluation metric:

$$\text{Macro F1} = \frac{1}{C}\sum_{c=1}^{C}\text{F1}_c \tag{1}$$

where C is the number of classes and F1_c is the F1-score for class c. This metric gives equal weight to each class regardless of frequency, making it particularly appropriate for imbalanced datasets.

3.5. Data Preparation

We create an 85/15 stratified train/validation split (random seed 42) to tune hyperparameters and ensemble weights. Stratification ensures that class proportions are preserved in both splits. All images are converted to RGB format for consistency. We do not use any external data or pre-training beyond the provided training set and the base Qwen3-VL model.

4. Methodology

Our system follows a two-stage training pipeline adapted from the RA-HMD framework (Mei et al., 2025). Figure 1 illustrates the overall architecture.

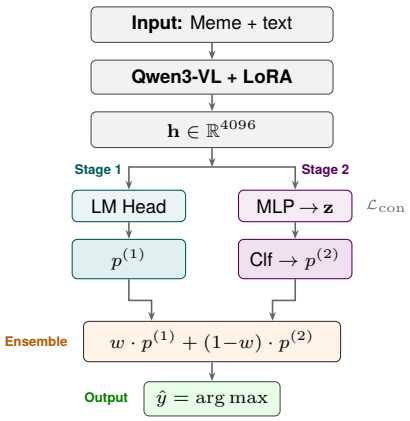


Figure 1: Architecture overview. Stage 1 uses the LM head for token probabilities. Stage 2 adds an MLP projection with contrastive loss ($\mathcal{L}_{\text{con}}$) and a classifier. Outputs are ensembled and the final prediction is made via argmax.

4.1. Base Model Selection

We select Qwen3-VL-8B-Instruct as our backbone for several reasons:

- **Native Devanagari support**: The model can read and understand text in Devanagari script directly from images, eliminating the need for separate OCR.

- **Strong multilingual capabilities**: Pre-training on diverse multilingual data enables transfer to low-resource languages like Nepali.
- **Dynamic resolution**: The model handles images of varying sizes (min: 256×28^2, max: 512×28^2 pixels) without fixed resizing.
- **Instruction following**: The model can follow complex task instructions, enabling prompt-based classification.

4.2. Stage 1: Generative Fine-Tuning with LoRA

In Stage 1, we fine-tune the model to generate class labels directly from meme images. This generative approach leverages the model's pre-trained language understanding capabilities.

4.2.1. Low-Rank Adaptation (LoRA)

Rather than fine-tuning all parameters, we apply LoRA (Hu et al., 2022) to inject trainable low-rank matrices into the model. For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, LoRA adds:

$$W = W_0 + \Delta W = W_0 + BA \tag{2}$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and $r \ll \min(d, k)$ is the rank. The scaling factor α/r modulates the contribution of the adaptation. Table 1 shows task-specific LoRA configurations.

Parameter	Hate Speech	Sentiment
Rank (r)	64	32
Alpha (α)	128	64
Scaling (α/r)	2.0	2.0
Dropout	0.05	0.1
Trainable params	$\sim$0.5%	$\sim$0.25%

Table 1: LoRA configurations. Sentiment uses smaller rank and higher dropout to prevent overfitting on the three-class problem.

4.2.2. Task-Specific Prompts

We design prompts that establish the model's role and provide clear classification criteria.

Hate Speech Detection Prompts:

System prompt: *"You are an expert at detecting hate speech in Nepali memes. Analyze both the visual content and any Nepali/Devanagari text in the image. Consider cultural context, targeted groups (ethnicity, religion, gender, caste), and implicit meanings."*

User prompt: *"Is this Nepali meme hateful?"*

Categories:

- *Hate: Content that targets or attacks groups based on ethnicity, religion, gender, caste, or promotes violence/discrimination*
- *Safe: Non-hateful content (neutral, positive, jokes without targeting groups)*

Answer with one word only: Hate or Safe.
Answer:

Sentiment Analysis Prompts:

System prompt: *"You are an expert at analyzing sentiment in Nepali memes and social media content. You understand Nepali culture, humor, sarcasm, and emotional expression. Analyze both the visual elements (images, facial expressions, symbols) and Devanagari/English text. Pay attention to cultural references, meme templates, and contextual meaning."*

User prompt: *"Analyze the overall sentiment of this Nepali meme."*

Categories:

- *Negative: Expresses criticism, mockery, sadness, anger, frustration, disappointment, or pessimism*
- *Neutral: Factual information, observations, or content without clear emotional valence*
- *Positive: Expresses joy, humor, encouragement, celebration, love, hope, or optimism*

Consider: (1) Visual content and facial expressions, (2) Text meaning (literal and contextual), (3) Cultural context and references, (4) Overall emotional tone.

Respond with ONLY one word: Negative, Neutral, or Positive.

Answer:

4.2.3. MLP Projection Head

Alongside LoRA, we train an MLP projection head that maps the model's last hidden state to a lower-dimensional embedding space:

$$\mathbf{h}_1 = \text{GELU}(W_1\mathbf{h} + b_1) \tag{3}$$

$$\mathbf{h}_2 = \text{LayerNorm}(\mathbf{h}_1) \tag{4}$$

$$\mathbf{z} = W_2\mathbf{h}_2 + b_2 \tag{5}$$

where $\mathbf{h} \in \mathbb{R}^{4096}$ is the last hidden state, $W_1 \in \mathbb{R}^{2048 \times 4096}$, $W_2 \in \mathbb{R}^{512 \times 2048}$, and $\mathbf{z} \in \mathbb{R}^{512}$ is the projected embedding.

This projection head serves two purposes: (1) dimensionality reduction for efficient contrastive learning in Stage 2, and (2) learning task-specific representations beyond what the language model head captures.

4.2.4. Training Objective

The Stage 1 training objective is the standard language modeling loss, but computed only over the target tokens. Given a prompt x and target label y (e.g., "Hate" or "Safe"), we minimize:

$$\mathcal{L}_{\text{LM}} = -\log P(y|x, \text{image}) \tag{6}$$

We mask the prompt tokens (set labels to -100) so that gradients flow only through the answer tokens. This focuses learning on the classification decision.

4.2.5. Class Imbalance Handling

Both datasets exhibit significant class imbalance. We employ three complementary strategies:

Minority Class Oversampling: We duplicate minority class samples to achieve balanced class frequencies. For hate speech, we target a 1:1 ratio (hate:safe). For sentiment, we target 1:1:1 (negative:neutral:positive). The majority class is not downsampled to preserve all available information.

Image Augmentation: Oversampled (duplicated) images undergo random transformations to increase diversity and prevent the model from memorizing specific samples. With 80% probability, we apply:

- Horizontal flip: 50% probability
- Rotation: uniform in $[-15°, +15°]$
- Brightness adjustment: factor in $[0.8, 1.2]$
- Contrast adjustment: factor in $[0.8, 1.2]$
- Saturation adjustment: factor in $[0.8, 1.2]$

Each transformation is applied independently with 50% probability, creating diverse augmented versions.

Focal Loss (Sentiment Only): For the three-class sentiment task, we additionally use focal loss (Lin et al., 2017):

$$\mathcal{L}_{\text{focal}} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{7}$$

with focusing parameter $\gamma = 2.0$. The modulating factor $(1 - p_t)^{\gamma}$ reduces the loss contribution from easy examples, focusing training on hard cases. Class weights α_t are set to inverse class frequencies.

4.2.6. Training Configuration

The sentiment task uses a higher learning rate and larger effective batch size. We found this necessary for the more complex three-class decision boundary.

Parameter	Hate Speech	Sentiment
Learning rate	5×10^{-5}	1×10^{-4}
Batch size	2	2
Gradient accum.	8	12
Effective batch	16	24
Epochs	5	5
Optimizer	AdamW (β_1=0.9, β_2=0.999)	
Weight decay	0.01	
Epsilon	10^{-8}	
LR scheduler	Cosine, 10% warmup	
Grad. clipping	1.0	
Precision	bfloat16 mixed	

Table 2: Stage 1 training configurations.

4.3. Stage 2: Contrastive Backbone Fine-Tuning

Stage 2 continues training the LoRA layers and MLP with a joint classification and contrastive objective. The goal is to learn embeddings where same-class samples cluster together while different-class samples are pushed apart.

4.3.1. Linear Classifier

We add a linear classifier on the projected embeddings:

$$\mathbf{y} = W_c \mathbf{z} + b_c \tag{8}$$

where $W_c \in \mathbb{R}^{C \times 512}$, $b_c \in \mathbb{R}^C$, and C is the number of classes (2 for hate speech, 3 for sentiment).

4.3.2. Supervised Contrastive Loss

We use a supervised variant of InfoNCE loss (van den Oord et al., 2018; Khosla et al., 2020). For each anchor embedding $\mathbf{z}_i$ with label y_i, we sample k^+ positive embeddings (same label) and k^- negative embeddings (different labels) from the training set.

The contrastive loss is:

$$\mathcal{L}_{\text{con}} = -\frac{1}{k^+} \sum_{j=1}^{k^+} \log \frac{\exp(s_{ij}^+ / \tau)}{D_i} \tag{9}$$

where $s_{ij}^+ = \text{sim}(\mathbf{z}_i, \mathbf{z}_j^+)$ is the cosine similarity between anchor i and positive j, and the denominator is:

$$D_i = \sum_{p=1}^{k^+} \exp(s_{ip}^+ / \tau) + \sum_{n=1}^{k^-} \exp(s_{in}^- / \tau) \tag{10}$$

where $\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b} / (\|\mathbf{a}\| \|\mathbf{b}\|)$ is cosine similarity and $\tau = 0.07$ is the temperature parameter. All embeddings are L2-normalized before computing similarity.

The temperature τ controls the concentration of the distribution. Lower temperatures make the

model more sensitive to hard negatives but can lead to training instability. We use $\tau = 0.07$ following prior work (Khosla et al., 2020).

4.3.3. Joint Training Objective

The Stage 2 loss combines classification and contrastive terms:

$$\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}} \tag{11}$$

For hate speech: $\lambda_{\text{cls}} = 1.0$, $\lambda_{\text{con}} = 0.3$. The classification loss uses label smoothing of 0.05 to prevent overconfident predictions.

For sentiment: $\lambda_{\text{cls}} = 1.0$, $\lambda_{\text{con}} = 0.4$. The classification loss uses inverse-frequency class weights to handle imbalance.

4.3.4. Stage 2 Training Configuration

Table 3 shows Stage 2 hyperparameters. A key difference from Stage 1 is the much lower backbone learning rate and stronger regularization, especially for sentiment.

Parameter	Hate Speech	Sentiment
Backbone LR	2×10^{-5}	5×10^{-6}
Classifier LR	1×10^{-4}	2×10^{-5}
Batch size	2	2
Gradient accum.	8	12
Epochs	5	3
Weight decay	0.01	0.1
Early stopping	–	2
Positives (k^+)	3	4
Negatives (k^-)	5	6
Temperature (τ)	0.07	0.07
λ_{con}	0.3	0.4
Label smoothing	0.05	–
Class weights	–	Inverse freq.

Table 3: Stage 2 training configurations. Sentiment uses aggressive regularization to prevent overfitting.

The sentiment task required substantially different settings:

- 4× lower backbone learning rate (5×10^{-6} vs 2×10^{-5})
- 10× higher weight decay (0.1 vs 0.01)
- Fewer epochs (3 vs 5) with early stopping

These aggressive regularization choices were necessary because the three-class sentiment task has a smaller effective training set per class and showed signs of overfitting early in our experiments.

4.4. Inference and Ensemble

At inference time, we combine predictions from both stages to leverage their complementary strengths.

4.4.1. Stage 1 Predictions

We extract token-level probabilities from the language model head. For hate speech:

$$p^{(1)}_{\text{hate}} = \frac{\exp(l_{\text{Hate}})}{\exp(l_{\text{Hate}}) + \exp(l_{\text{Safe}})} \tag{12}$$

where l_{Hate} and l_{Safe} are logits for the respective tokens at the first generated position.

For sentiment, we similarly compute:

$$p^{(1)}_c = \frac{\exp(l_c)}{\sum_{c'} \exp(l_{c'})} \tag{13}$$

for each class $c \in \{\text{Negative}, \text{Neutral}, \text{Positive}\}$.

4.4.2. Stage 2 Predictions

We apply softmax to the linear classifier outputs:

$$p^{(2)} = \text{softmax}(W_c\mathbf{z} + b_c) \tag{14}$$

4.4.3. Ensemble Strategy

The final prediction is a weighted combination:

$$p_{\text{ensemble}} = w \cdot p^{(1)} + (1 - w) \cdot p^{(2)} \tag{15}$$

We tune the ensemble weight w and decision threshold jointly via grid search on the validation set, optimizing for macro F1. For hate speech:

- Ensemble weight w: search in $[0.1, 0.9]$ with step 0.02
- Decision threshold: search in $[0.25, 0.75]$ with step 0.005

For sentiment (multi-class), we use argmax on the ensembled probabilities and only tune the ensemble weight.

5. Experimental Setup

5.1. Implementation Details

We implement our system using:

- PyTorch 2.4.0 with CUDA
- HuggingFace Transformers $\geq$ 4.46.0 (Wolf et al., 2020)
- PEFT $\geq$ 0.13.0 for LoRA (Mangrulkar et al., 2024)
- Accelerate $\geq$ 1.0.0 for distributed training

Training runs on NVIDIA H100 GPUs via Modal Labs cloud infrastructure. Each training run takes approximately 2–4 hours per stage.

5.2. Reproducibility

We use fixed random seeds (42) for data splitting, model initialization, and training. All hyperparameters are specified in Tables 1–3. Our code will be made available upon publication.

6. Results and Discussion

6.1. Official Results

Subtask	Macro F1	Rank
A: Hate Speech Detection	0.7970	2nd
B: Sentiment Analysis	0.5177	4th

Table 4: Official leaderboard rankings on CHiPSAL 2026.

Table 4 shows our official results. We achieved strong performance on hate speech detection with F1 of 0.797, securing 2nd place. For sentiment analysis, we achieved F1 of 0.518, placing 4th. The binary hate speech task proved more tractable, likely due to the clearer class boundary compared to the three-way sentiment distinction.

6.2. Ablation Study

Configuration	Test F1
Subtask A: Hate Speech Detection	
Stage 1 (LoRA baseline)	0.74
+ Oversampling	0.76
+ Image augmentation	0.77
+ Stage 2 contrastive	0.79
+ Threshold tuning	**0.80**
Subtask B: Sentiment Analysis	
Stage 1 (LoRA baseline)	0.49
+ Focal loss	0.50
+ Oversampling & augmentation	0.51
+ Stage 2 contrastive	**0.52**

Table 5: Ablation study. Each row adds one component cumulatively. F1 scores are reported on the held-out test set, except threshold tuning which was optimized on the validation split and then applied to the test set.

Table 5 presents detailed ablation results. Key findings:

LoRA scope matters. Applying LoRA to all attention and feed-forward layers outperformed applying it only to attention layers (+3 points on hate speech).

Class imbalance handling is crucial. The combination of oversampling and augmentation contributes 2–3 points. Focal loss adds another point for the three-class sentiment task by addressing neutral class bias.

Contrastive learning provides consistent gains. Stage 2 adds 2 points on hate speech and contributes to sentiment, demonstrating that the contrastive objective learns more discriminative representations.

Threshold tuning improves hate speech detection. Optimizing the classification threshold on the validation split and applying it to the test set adds 1 point, reaching the final score of 0.80.

6.3. Qualitative Examples

Successful Cases: The model correctly identifies explicit hate speech with visual symbols (e.g., derogatory depictions) combined with text. For sentiment, clear emotional expressions in both image and text (e.g., smiling faces with celebratory text) are reliably classified.

Failure Cases: A meme showing a historical figure with a quote that is hateful in Nepali cultural context but neutral in isolation was misclassified as safe. The model lacks sufficient cultural knowledge to interpret such references.

7. Conclusion

We presented a two-stage vision-language system for Nepali meme classification at CHiPSAL 2026. By combining LoRA fine-tuning with contrastive learning on Qwen3-VL-8B-Instruct, we achieved 2nd place on hate speech detection (F1: 0.797) and 4th place on sentiment analysis (F1: 0.518).

Our results highlight several key findings. Modern VLMs with native Devanagari support enable end-to-end processing of South Asian language memes without OCR pipelines, eliminating error propagation from intermediate steps. Two-stage training (generative then contrastive) outperforms either approach alone by 3–5 F1 points, demonstrating the complementarity of language modeling and metric learning objectives. Task-specific hyperparameter tuning proved essential, as sentiment required much stronger regularization than hate speech. Finally, careful handling of class imbalance is crucial for macro F1 optimization, contributing up to 7 points across our ablation.

Promising future directions include retrieval-augmented approaches to incorporate cultural knowledge, synthetic data augmentation to expand limited training sets, multi-task learning jointly across hate speech and sentiment, and explainability analysis to better understand model decisions for culturally sensitive content.

8. Limitations

- **Data scarcity:** With approximately 1,000 samples per task, overfitting remains a concern. Our validation results may not fully generalize to the

test distribution. Due to the high computational cost of each training run, we did not evaluate variance across multiple random seeds; results may vary with different initializations.

- **Cultural context:** Nepali cultural references, historical context, and humor styles may not be well-represented in Qwen3-VL's pretraining data, limiting performance on culturally-specific content.
- **Prompt sensitivity:** We did not exhaustively search prompt variations, which could further improve performance.
- **Code-mixing:** While Qwen3-VL handles Nepali-English code-mixing to some extent, highly mixed content may still pose challenges.
- **Baseline comparisons:** We did not evaluate Qwen3-VL-8B-Instruct in a zero-shot (no fine-tuning) setting or compare against an OCR + text-only pipeline, which would better quantify the contribution of our end-to-end multimodal approach. We leave these comparisons for future work.

9. Ethical Considerations

Hate speech detection systems must be deployed carefully to avoid over-censorship or bias. Our model may have false positives on legitimate political speech or cultural content unfamiliar to the base model. We recommend human review for content moderation decisions.

10. Acknowledgements

We thank the CHiPSAL 2026 organizers for creating this valuable benchmark and shared task infrastructure. We acknowledge Modal Labs for computational resources.

11. Bibliographical References

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673.

Douwe Kiela, Hamed Firooz, Aravind Mober, Vedanuj Goswami, Amanpreet Jambhekar, Patrick Hwang, Arthur Kiela, Holger Schwenk, et al. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2024. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Jingyu Mei, Jingyuan Chen, Weihua Lin, Bill Byrne, and Marcus Tomalin. 2024. Improving hateful meme detection through retrieval-guided contrastive learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5333–5347.

Jingyu Mei, Jingyuan Chen, Guang Yang, Weihua Lin, and Bill Byrne. 2025. Robust adaptation of large multimodal models for retrieval augmented hateful meme detection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 23817–23839.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. MOMENTA: A multimodal framework for detecting harmful memes

and their targets. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Surabhi Adhikari, Hariram Veeramani, Mehwish Nasim, and Usman Naseem. 2025a. Large language models (llm) in computational social science: prospects, current state, and challenges. *Social Network Analysis and Mining*, 15(1):1–30.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025b. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025c. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Team Oryu@CHiPSAL 2026:Integrating Text and Vision Transformers for Multimodal Hate Speech Detection in Memes

Noore Tamanna Orny, Joyeta Barua Moni, Md. Abtahee Kabir, Hasan Murad
Chittagong University of Engineering and Technology
Pahartali, Raozan, Chattogram, Bangladesh
{u2104092, u2104130, u2104089}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

With the proliferation of multimodal content on various social media platforms, automated hate speech detection has emerged as a challenge, especially in meme-based communication, where meaning arises from interactions between text and images. In these situations, unimodal techniques are inadequate in capturing semantics. In order to address such issues, a late-fusion-based multimodal hate speech detection framework has been proposed and implemented for the CHiPSAL shared task. In the proposed framework, multimodal content is processed by utilizing XLM-RoBERTa for multilingual text representation and a Vision Transformer (ViT) for visual representation. Both modal representations are fused using a fully connected classification head and are used for binary hate speech detection. The findings suggest that multimodal content effectively captures features from individual modalities and helps improve hate speech detection accuracy by obtaining a Macro F1-score of 0.66 and ranking 5th on the leaderboard. Also, transformer-based multimodal fusion performs effectively and acts as a reliable baseline for hate speech detection in low-resource multilingual meme-based communication scenarios.

Keywords: Multimodal Hate Speech Detection, Meme Analysis, Vision Transformer (ViT), XLM-RoBERTa, Multimodal Learning, Hate Meme Classification, Social Media Moderation, Macro F1 Optimization, Transformer-based Models, Multilingual NLP

1. Introduction

With the proliferation of multimodal content on various social media platforms, automated hate speech detection has emerged as a challenge, especially in meme-based communication, where meaning arises from interactions between text and images. In such scenarios, unimodal-based approaches have been found inadequate in capturing semantics from multiple modalities. In order to address such issues, a late-fusion-based multimodal hate speech detection framework was proposed and implemented for the CHiPSAL shared task (Thapa et al., 2026; Sarveswaran et al., 2026). In the proposed framework, multimodal content is processed by utilizing XLM-RoBERTa for multilingual text representation and a ViT for visual representation. Both modal representations are fused using a fully connected classification head and are used for binary hate speech detection.The major contributions of this work can be summarized as follows:

1. This work presents a transformer-based multimodal hate speech detection system specifically designed to handle the complexities of low-resource multilingual memes.

2. This work validates the effectiveness of the XLM-RoBERTa and Vision Transformer architectures in the context of the hate speech detection task, specifically through the application of the late-fusion strategy.

3. This work presents the application of class-weighted training, which is effective in reducing the issue of class imbalance for this task. This work presents competitive performance securing the 5th position on the leaderboard.

While the proposed approach is an improvement over existing transformer-based architectures, the major contribution is in the systematic analysis of the effectiveness of late fusion approaches, class imbalance handling, and the integration of multimodal features in a low-resource code-mixed meme environment. This study offers significant empirical evidence of the effectiveness of the proposed approaches and hence serves as a robust baseline for future studies.

2. Related Work

Due to the rapid growth of user-generated content on social media, there is a considerable increase in harmful and hateful content, which led to extensive research on automated hate speech detection systems. Initial research on this topic was mostly focused on text-based classification problems using traditional machine learning techniques, including Support Vector Machines and logistic regression models with handcrafted features. Although these models demonstrated considerable success, they failed to capture contextual nuances, sarcasm, and implicit hate speech, which are often encountered in real-world datasets (Maharjan et al., 2025).

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 284–291
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

Early studies have also dealt with the challenges that are encountered during the detection of hate speeches on social media platforms, particularly when using automated detection techniques. The research in this area has dealt with the detection of hate speeches on social media platforms with the help of natural language processing techniques. The various challenges that are encountered during the detection of hate speeches on social media platforms with the help of NLP-based detection mechanisms include contextual ambiguity, sarcasm, domain variability, and the difficulty of identifying implicit hateful expressions that are common in online discourse. These complex linguistic features are difficult to detect with the help of NLP-based detection mechanisms. Therefore, advanced detection mechanisms, such as deep learning, for hate speeches on social media platforms are required (Parihar et al., 2021).

Recently, with the introduction of deep learning techniques, transformer-based language models including BERT and its multilingual variants have demonstrated considerable success in improving the performance of hate speech detection systems. Multilingual variants, including XLM-RoBERTa, have demonstrated considerable potential for handling low-resource languages by applying cross-lingual transfer learning techniques. These models are able to learn contextualized word representations, which are more accurate than traditional word embeddings for semantic meaning representation (Das et al., 2024). However, these text-based systems are insufficient for meme-based analysis, as hate speech is often represented by interacting with images. In order to address this drawback, there has been a growing tendency towards multimodal-based hate speech detection. This method uses a combination of text and image features to better comprehend hate speech, as the latter often makes use of images to convey its intended meaning. ViT and CNN-based models have emerged as popular choices for image encoding in multimodal-based hate speech detection methods (Thapa et al., 2025a). These models facilitate the effective extraction of semantic features from images.

The process of multimodal learning relies heavily on the fusion mechanism employed. Early fusion, late fusion, and cross-modal attention-based methods have been explored in previous studies. It has been demonstrated that late fusion-based methods, which involve the independent encoding of each modality followed by their combination, can offer a robust yet computationally efficient solution for multimodal learning tasks (Qi and Wei, 2025). Recent studies have introduced the concept of gated fusion and attention-based methods to weigh the relative importance of each modality, thereby improving the robustness of the model in the presence of noisy or misleading modalities(Piot and Parapar, 2025)

Upon these developments, the present study employs a late fusion multimodal transformer model with integrated XLM-RoBERTa for text understanding and ViT for feature extraction from images. The motivation is to capitalize on the strengths offered by pre-trained multimodal representations while ensuring stability and efficiency for meme classification in resource-scarce environments.

3. Data Description

3.1. Dataset Overview

The experiments of this work were performed on a multimodal Nepali meme dataset that was created to perform hate speech detection (Thapa et al., 2025b,c). Each instance of the meme in the dataset contains both textual and visual information, representing the multimodal nature of internet memes. The problem is defined as a binary classification problem where the system needs to predict whether the meme is hateful or non-hateful.

Each instance of the meme contains information related to the image of the meme, textual information related to the meme, and a label related to the meme. The dataset is challenging since the hateful information is encoded based on the interaction of both textual and visual information of the meme. The annotation framework used in the dataset follows the multimodal hate speech annotation guidelines introduced in the CrisisHateMM dataset (Bhandari et al., 2023).

Label	Training	Evaluation	Test
Hate (1)	720	98	–
Non-hate (0)	348	35	–
Total	**1068**	**133**	**134**

Table 1: Label-wise Distribution of Training, Evaluation, and Test Data

3.2. Data Splits

The dataset is split into training, validation(evaluation) and test sets.The training split contains 1,068 samples, the validation(evaluation) split contains 133 samples, and the test split contains 134 samples. In the validation set, the hate class includes 98 samples and the non hate class includes 35 samples, which indicates a noticeable imbalance like the training set. The evaluation dataset was used only to perform model selection, monitoring of hyperparameters, and initial model performance evaluation, while the test data was used only to perform the final submission and leaderboard evaluation.

3.3. Label Distribution

From the dataset, we can see that there is a class imbalance issue, as the hate class dominates the dataset. In the training set, the hate class(label=1) has 720 instances, whereas the non-hate class(label=0), has only 348 instances. In the evaluation set, there are 98 hate class instances, whereas there are only 35 instances of the non-hate class.

3.4. Text Characteristics

The text data portion of the given data set includes small meme captions, as well as other textual data that is full of noisy, contextual, and informal language. The text data is full of slang, spelling mistakes, code-switching, emojis, and other contextual expressions, making it more complicated for the model to detect hate text automatically. The text data also includes many implicit expressions of hateful intent, such as sarcasm, humor, etc., rather than explicit hateful words or phrases. The text data portion is also too short, as it is part of the meme, making it more complicated for the model to classify the text data correctly.

3.5. Image Characteristics

The visual component of the dataset comprises meme images sourced from social media platforms, which have varying levels of diversity in their visual content and the nature of the image and content used in the memes. These images have varying resolutions and ratios, ranging from low-resolution compressed images to more visually appealing and professionally created images, although in the interest of model compatibility, all the images have been preprocessed to 224x224 pixels. These images have varying levels of text overlay, symbolization, and graphical content, which are visually representative of the memes and the context in which the memes are used. Additionally, the dataset has images that are natural and those that have been artificially created, including complex backgrounds and varying levels of color distribution in the images.

3.6. Preprocessing

For textual processing, meme text was tokenized using the XLM-RoBERTa tokenizer with a maximum sequence length of 128 tokens. For the visual modality, images were resized to 224×224 pixels, randomly horizontally flipped during training for data augmentation, normalized, and converted to tensor format. These preprocessing steps ensure compatibility with the pretrained transformer encoders while improving model generalization. As shown in Figure 1, the proposed framework consists of modality-specific encoders followed by a multimodal fusion module.

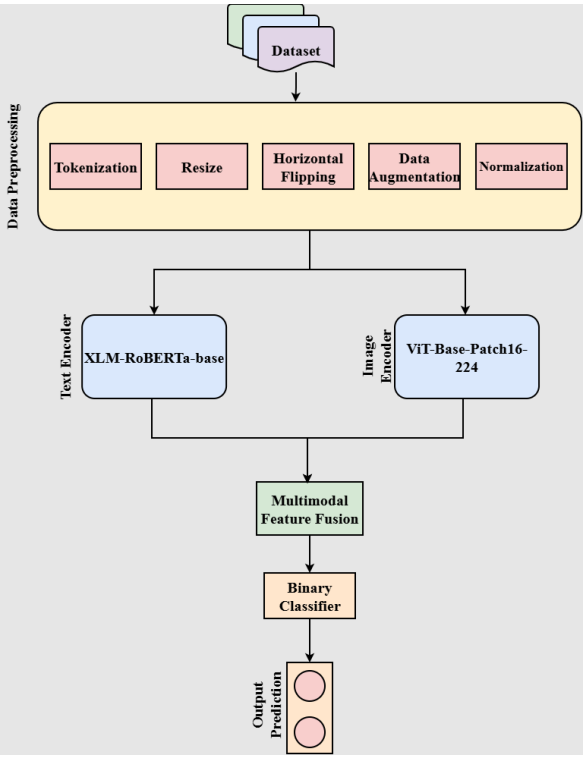


Figure 1: Proposed multimodal hate speech detection architecture.

4. Methodology

4.1. Task Definition

The problem this paper aims to solve is the multimodal hate speech detection in memes. In this case, the information available in the memes will be both textual and visual.

The problem can be formally defined as follows: given a meme with text information T and image information I, the aim of the model will be to learn the following function-

$$f(T, I) \rightarrow y$$

where y $\in$ (0 , 1) corresponds to hate/ non-hate.

4.2. Overall Architecture

The proposed system utilizes a late fusion multimodal transformer with four major components:

1. Multilingual Text Encoder: XLM-RoBERTa
2. Vision Transformer Image Encoder: ViT
3. Feature Fusion Module

4. Fully Connected Classification Head

This architecture utilizes modality-specific encoders and fuse the features using a concatenation method followed by a non-linear transformation.

4.3. Text Encoding

For textual semantics, we use the pre-trained model XLM-RoBERTa-base(Conneau et al., 2019), which is a transformer model pre-trained on cross-lingual large-scale data. This model has been chosen due to the multilingual content of the meme text in the dataset, which includes Nepali, English, and code-switched text.

The input sequence T, the text embedding of which we want to find, is first split into subword units using the tokenizer. Then the transformer model processes the sequence to find the text embedding

$$h_T = \mathrm{XLM{-}R}(T)$$

The text embedding has a dimensionality of 768 units.

4.4. Image Encoding

For visualization purposes, we will use Vision Transformer (ViT-Base-Patch16-224)(Wu et al., 2020) which has been pre-trained on ImageNet. Unlike convolutional neural networks, ViT uses a self-attention mechanism to consider global relationships in images.

Each image is resized to 224x224 pixels. This image is divided into 16×16 patches. The patches are then embedded and passed through the transformer encoder to obtain a vision representation:

$$h_I = \mathrm{ViT}(I)$$

The classification head is removed from the ViT. The 768-dimensional vector is used as the image embedding. ViT is used because:memes require global context understanding, hateful cues are spatially distributed, transformers are similar to text transformers

4.5. Multimodal Fusion

After getting the modality-specific features, the model applies late fusion using feature concatenation. The text and image embeddings are concatenated in the feature dimension which is a 1536-dimensional representation.

$$h_F = [h_T; h_I]$$

The late fusion approach is used because: pre-trained encoders can be utilized separately,it is robust with limited training data, it alleviates overfitting across modalities

4.6. Classification Head

The fused representation is then passed through a fully connected neural network to capture cross-modal interactions. The classification head is composed of the following layers:

Component	Configuration
Dropout	0.3
Linear Projection	$1536 \rightarrow 512$
Activation Function	ReLU
Dropout	0.3
Linear Layer	$512 \rightarrow 2$

Table 2: Classification Head Architecture

The mathematical representation of the classification head is given by:

$z = ReLU(W_1\ \mathsf{h}_F + \mathsf{b}_1\ ),\ \hat{y} = W_2 \mathsf{z} + \mathsf{b}_2$

The last layer generates logits for the binary hate classification problem.

This was done using the classification head as dimensionality reduction improves generalization, the ReLU(Nair and Hinton, 2010) activation function adds non-linearity to the cross-modal learning, and the dropout adds regularization to prevent overfitting as the dataset used was not very large.

4.7. Handling Class Imbalance

The dataset has a class imbalance problem, where one class has significantly more instances than the other. To tackle this issue, a class-weighted cross-entropy loss is used. Class weights are calculated using the following formula based on the balanced heuristic:

$$w_c = \frac{N}{K \cdot N_c} \tag{1}$$

where N_c is the number of instances in the class c.

The weighted loss is calculated as follows:

$$\mathcal{L} = -w_y \log p_y$$

As we address the class imbalance, it-

1. Improves minority class recall.
2. Stabilizes macro-F1.
3. Prevents majority bias.

4.8. Training Procedure

The proposed multimodal model was trained using an end-to-end approach to perform three epochs of training. In the case of each meme, the input text was tokenized, and the contextual text embedding was obtained using the pre-trained XLM-RoBERTa model, whereas the input image was passed through the ViT model to extract the visual

feature embedding. The obtained feature embedding from both the image and the text was concatenated to form a single feature vector, which was passed through a fully connected classification head to obtain the hate and non-hate class logits.

The model was optimized using the AdamW(Loshchilov and Hutter, 2019) optimizer with a learning rate of 2e-5, whereas class-weighted cross-entropy loss was used to handle the class imbalance issue, and the model selection was carried out using the best model based on the validation set's Macro F1-score.

4.9. Implementation Details

The model was developed using PyTorch as a deep learning library, as well as Hugging Face's Transformers library for transformer models. Text features were represented using a pretrained XLM-RoBERTa-base model with a sequence length of 128 tokens. Visual features were represented using a pretrained Vision Transformer (ViT-Base-Patch16-224), which was accessed via the timm library, resizing images to 224x224 pixels and randomly flipping images horizontally. A fully connected fusion network was used to fuse the 768-dimensional text and image embeddings, with dropout set to 0.3 for a binary classification problem. AdamW was used as the optimizer with a learning rate of 2×10^{-5},as well as class-weighted cross-entropy loss to account for class imbalance. Three epochs were trained with a batch size of 4, and the model was selected based on Macro F1-Score on the evaluation set.The training configuration of the model has been shown in Table 3:

Parameter	Value
Framework	PyTorch
Text Model	XLM-RoBERTa-base
Image Model	ViT-Base-Patch16-224
Image Size	224×224
Max Text Length	128
Optimizer	AdamW
Learning Rate	2×10^{-5}
Dropout	0.3
Epochs	3
Batch Size	4
Hardware	NVIDIA GPU (Kaggle)

Table 3: Training Configuration

5. Results and Discussion

5.1. Evaluation Metric

The multimodal model was evaluated based on the official CHiPSAL shared task protocol. The main evaluation criteria used were Macro F1-score, which is mostly used for imbalanced classification problems since all classes are given equal weightage. In addition to Macro F1-score, Accuracy, Macro Precision, and Macro Recall were used to keep track of the model's performance.

Accuracy is defined as the proportion of samples that were predicted and actually belonged to the class predicted by the model.

Macro Precision is defined as the proportion of positive class predictions made by the model.

Macro Recall evaluates how well the model is able to recall positive samples across all classes. The Macro F1-score is the harmonic mean of Precision and Recall computed separately for each class and then averaged over all classes.

5.2. Quantitative Results

The performance of the proposed late fusion multimodal model on the test set is shown in Table 4.

Metric	Score
Macro F1	0.66
Accuracy	0.68
Precision (Macro)	0.64
Recall (Macro)	0.55

Table 4: Performance of the Proposed Model

To comprehend the contribution of each mode better, we also test the performance of the unimodal models which is shown in Table 5.

Model	Macro F1
Text-only (XLM-RoBERTa)	0.21
Image-only (ViT)	0.45
Proposed	0.52

Table 5: Comparison of Unimodal and Proposed Models using Validation Set

In this case, the text-only model uses XLM-RoBERTa, while the image-only model uses the Vision Transformer (ViT). The results show that the multimodal model has a better performance than the other models, illustrating the effectiveness of the fusion of textual and visual modalities for meme comprehension. All the approaches in Table 5 are conducted under consistent experimental settings, using the same dataset, trained for 3 epochs with a learning rate of 2×10^{-5}, and the same loss function. On the official Codabench leaderboard, our system has achieved: Macro F1: 0.66 , Leaderboard Rank: 5th place

5.3. Analysis of Model Performance

The performance of the multimodal hate speech detection system was enhanced gradually with the improvement of the model architecture and the training strategy. Initially, the multimodal baseline model achieved a Macro F1-score of 0.40, which implied that the model was not effective enough to handle the relationship between the hate speech and the visual content of the memes. The model was also challenged to make balanced decisions on the hate speech classification.

A gated fusion mechanism was incorporated to address the challenges associated with the initial multimodal model. With the enhanced model, the Macro F1-score was improved to 0.52. This was due to the incorporation of the gating mechanism that allowed the model to control the amount of information to be used from the textual and visual content of the hate speech memes. The model was improved to be effective in highlighting the importance of the modality of the hate speech content. The enhanced model was also challenged to be somewhat sensitive to noise and instability. Significantly high performance gain was achieved using the final late fusion transformer model, which attained a Macro F1-Score of 0.66. Several reasons contributed to the performance gain. First, the application of class-weighted cross-entropy loss effectively addressed the class imbalance problem, allowing the model to learn more balanced decision boundaries between the hate and non-hate classes. Second, the late fusion approach ensured greater stability in the model as the text encoder and image encoder learned strong modality-specific representations before fusion. Third, the effective handling of the data and the application of consistent preprocessing ensured the quality of the input data, which in turn ensured greater stability in the model.

The model attained its best performance within the first three epochs, which confirms the effectiveness of the pre-trained XLM-RoBERTa and Vision Transformer backends in the model. However, the slight variation in the performance metric across the epochs confirms that the model is still slightly sensitive to the presence of noisy data in the dataset and the class imbalance problem, which can be addressed in the future.

Model	Macro F1
Basic multimodal baseline	0.40
Gated fusion variant	0.52
Final late fusion transformer model	0.66

Table 6: Comparison of Model Variants

.

5.4. Error Analysis

Despite the strong performance in competitive scenarios, the model faces obstacles in a few challenging scenarios.

1. The model struggles with memes where hate is implied rather than stated. The reasons for this include a lack of deep cross-modal reasoning and insufficient world knowledge.

2. Sarcastic memes continue to be challenging due to the ambiguity of the text's sentiment and the image. The literal meaning of the image and the intended meaning of the meme can be very different.

3. The model uses the Vision Transformer for visual understanding, which inherently processes the textual patterns present in the images through patch embeddings. However, it does not explicitly include text extraction through dedicated mechanisms such as OCR. As a result, the model struggles with the hateful content present in the text of the images in the case of memes.

4. Regional memes can be challenging for the multilingual model, particularly when there is limited linguistic resource. As depicted in the confusion matrix in Figure 2, a substantial number of hate instances are correctly classified. Nevertheless, the system also experiences a considerable number of errors. To be precise, the system incorrectly classifies 26 instances of non-hate as hate, while it also incorrectly classifies 24 instances of hate as non-hate.

This also reveals that the system is highly sensitive to certain cues, either lexical or visual, which it perceives as indicative of hate. At the same time, it is also evident that certain hate instances are not as obvious for the model, which does not leverage multimodal reasoning.

Error Type	Count
False Positives	26
False Negatives	24

Table 7: Error Analysis on Validation Set

6. Conclusion

In this study, a multimodal architecture is proposed where text representation learning is accomplished using XLM-RoBERTa, while a Vision Transformer is used for image feature extraction. A late fusion mechanism is employed along with a fully connected classification network. To address class imbalance, class-weighted cross-entropy loss is

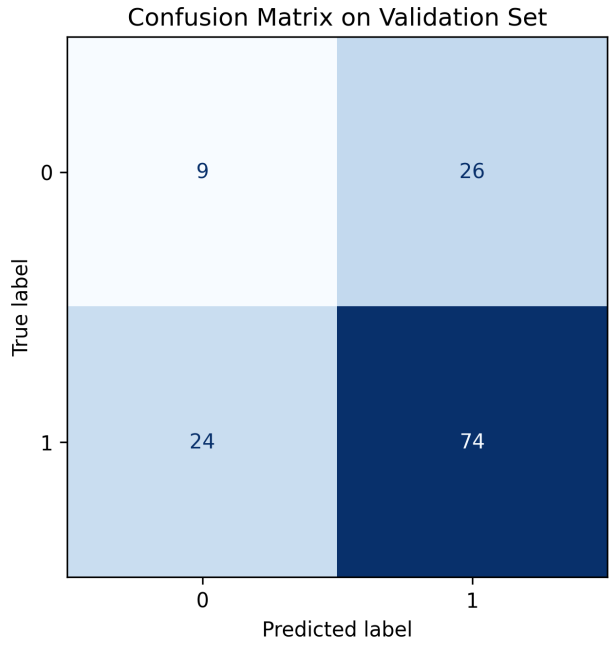


Figure 2: Confusion Matrix on Validation Set

incorporated, and image augmentation is applied to improve model performance. Based on the experimental results obtained using the CHiPSAL shared task dataset, the multimodal architecture effectively captures information from both modalities and maintains stable performance on the evaluation set. This demonstrates the effectiveness of transformer-based multimodal learning for meme hate speech detection while remaining computationally efficient through the late fusion design.

7. Limitations

Several avenues for further research exist for the proposed multimodal hate speech detection approach. First, the scope of the proposed work should be extended by incorporating more data for the training process, including more meme collections, for the detection of hate speech across different domains. Second, more sophisticated multimodal fusion methods, such as attention-based or region-token alignment, should be explored. Third, domain adaptive fine-tuning of the proposed pretrained transformers should be explored.

Furthermore, the proposed multimodal hate speech detection approach should be extended by incorporating more contextual information, such as user data, for the detection of implicit hate speech. Finally, more robust training methods should be explored for the proposed multimodal hate speech detection approach for the detection of hate speech across different domains as well as more efficient multimodal architectures.

Moreover, this work is centered on late fusion due to its stability and effectiveness in low-resource settings, alternative fusion strategies such as early fusion and cross-modal attention-based methods could further enhance interaction between modalities. Exploring these approaches may provide deeper multimodal alignment and is an important direction for future work.

8. Ethical Statement

In this research, a multimodal hate speech detection model is proposed based on the publicly available meme data from the shared task of the CHiPSAL dataset. Ethical considerations are also taken into account in the research. The data is publicly available and is curated for research purposes. There is no personal information available in the dataset. Therefore, the privacy of the data is not compromised in any way. To avoid the bias in the data available for the hate speech dataset, class-weighted cross-entropy loss is used in the model. The model is also pre-trained using the multilingual XLM RoBERTA model. However, the bias may still be present in the data because of the less diverse data available.The proposed system is strictly for research and decision support in content moderation and is not intended for fully autonomous use without human supervision because automated systems may also yield false positives or false negatives, which could impact the user. Data augmentation and balancing are also included in the research for fairness and generalization. However, detecting hate in multimodal content is difficult, especially for less represented cultures. The research follows the ethical principles of AI and the ethical guidelines set in the shared task. The model is used for classification of existing content only and does not produce content that is harmful.

9. Bibliographical References

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Susmita Das, Arpita Dutta, Kingshuk Roy, Abir Mondal, and Arnab Mukhopadhyay. 2024. A survey on automatic online hate speech detection in low-resource languages.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Sujal Maharjan, Astha Shrestha, Shuvam Thakur, and Rabin Thapa. 2025. Multimodal kathmandu@CASE 2025: Task-specific adaptation of multimodal transformers for hate, stance, and humor detection. In *Proceedings of the 8th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Texts*, pages 107–114, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA. Omnipress.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Paloma Piot and Javier Parapar. 2025. Towards efficient and explainable hate speech detection via model distillation. In *European Conference on Information Retrieval*, pages 376–392. Springer.

Meng Qi and Chung-Lun Wei. 2025. Lamha: Efficient multimodal malicious meme classification via lora-tuned adaptation and attentive-mlp fusion. In *Proceedings of the 2025 7th International Conference on Control and Computer Vision*, ICCCV '25, page 62–68, New York, NY, USA. Association for Computing Machinery.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Siddhant Bikram Shah, Kritesh Rauniyar, Shuvam Shiwakoti, Surabhi Adhikari, Hariram Veeramani, Kristina T. Johnson, Ali Hurriyetoglu, Hristo Tanev, and Usman Naseem. 2025a. Multimodal hate, humor, and stance event detection in marginalized sociopolitical movements. In *Proceedings of the 8th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Texts*, pages 20–31, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025b. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025c. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. 2020. Visual transformers: Token-based image representation and processing for computer vision.

Cuet Yet Another Baseline@CHiPSAL LREC 2026: Shared Task on Multimodal Sentiment Understanding in Low-Resource Memes

Rotna Dipika Debnath, Shahrin Afroz Hoque Ruhi, Ayesha Labiba, Arpita Mallik, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2104002, u2104015, u2104008, u2004023}@student.cuet.ac.bd
hasanmurad@cuet.ac.bd

Abstract

Memes serve as a method to express feelings such as humor, sarcasm, and diverse viewpoints. The task of identifying sentiment in memes is becoming increasingly complex, particularly in low-resource languages like Nepali where memes often combine images, texts, and code-mixed language. However, multimodal methods for sentiment analysis in Nepali memes seem to be insufficient. In this paper, we present our system for the Subtask B(Sentiment Analysis) for Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes@CHiPSAL LREC 2026. We implement various unimodal models, such as XLM-RoBERTa-large,MuRIL-base, Twitter-XLM-R for text. Moreover, we incorporate BLIP-2 captions to enhance visual-text understanding and adopted a multimodal approach that fuses textual embeddings, image embeddings, caption embeddings, and similarity scores. The fused features process through cross-attention and a dense neural network for classification, with focal loss and class weighting used to improve performance. Our approach achieved a macro F1 score of 0.50 securing 7th place and highlighting the importance of cross-modal interaction and large-scale pretrained vision-language models for robust meme understanding in sentiment analysis.

Keywords: Multimodal Sentiment Analysis, Low-Resource Languages, Nepali NLP, Vision–Language Models, Cross-Modal Attention,BLIP-2,CLIP,XLM-RoBERTa,Code-Mixed Text,Multimodal Fusion

1. Introduction

The concept of a "meme" was first introduced to describe genetic traits that are shaped by the sociocultural environment. Originally an academic term, this phrase was later adopted by users on the internet. A look at Google Trends indicates a significant increase in interest in the topic since 2011, with over 1.9 million search results for 'Internet meme', indicating their considerable cultural relevance (Shifman, 2013).

Memes are often shared with humorous or satirical intent and have emerged as a popular medium for expressing opinions and ideas. Analyzing memes is crucial as they convey sentiments and offers insights into public opinion, trends, and social dynamics (Nguyen et al., 2022). However, their growing popularity has also led to the spread of offensive and hateful content, posing challenges for content moderation (Parihar et al., 2021). Research has increasingly focused on analysis that considers both textual and visual elements. Nevertheless, most existing studies emphasize high-resource languages with limited work on low-resource languages such as Nepali. Despite the strong presence of politically and socially driven memes in Nepali online spaces, research on Nepali and code-mixed memes remains scarce, highlighting the need for more robust multimodal approaches in this context. Figure 1 shows examples of online memes that contain both text and image.

The purpose of this paper has been to perform sentiment analysis on Nepali memes. The CHiPSAL@ACL 2026 conference (Thapa et al., 2026) has introduced a dataset under Subtask B (Sentiment Analysis) of the Shared Task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes(Sarveswaran et al., 2026) which has consisted of multimodal memes annotated with sentiment labels as negative, neutral, and positive where each instance includes both textual content and an associated image, enabling the investigation of multimodal fusion strategies for sentiment understanding in a low-resource language setting.

To achieve our objective, we have designed a multimodal pipeline combining OCR-extracted text, BLIP-2 captions, CLIP ViT-L/14 image–text embeddings and XLM-RoBERTa-large for multilingual text. A cross-attention mechanism has enabled textual features to become image-aware, while focal loss with class weighting, ensemble learning, and test-time augmentation have improved robustness. Experimental results has shown that the proposed approach effectively captures subtle cross-modal cues to determine sentiment in Nepali memes. With an F1 score of 0.50, our approach has ranked 7th in the competition. The main contributions of this work have been:

- We have integrated XLM-RoBERTa, BLIP-2, and CLIP with cross-attention-based fusion in

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 292–299
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026



(a) Positive



(b) Negative



(c) Neutral

Figure 1: Examples of memes representing different sentiment classes.

a multimodal architecture.

- We have created a preprocessing pipeline that combines similarity-aware visual features, generated captions, and OCR-extracted text.
- For robust evaluation, we have used stratified 4-fold cross-validation with test-time augmentation and ensemble weighting.
- We have systematically investigated OCR noise filtering strategies for low-resource Nepali meme text, demonstrating that BLIP-2 visual captioning provides inherent robustness to OCR noise, making explicit filtering unnecessary.

2. Related Work

Several studies have addressed the difficulties of identifying sentiment and harmful content embedded within memes in the current research on multimodal meme understanding. Prior studies in this domain can be categorized based on their approach—text-based, image-based, and multimodal—as well as their use of machine learning, deep learning, and pre-trained models. Research efforts have focused on binary vs. multi-label classification and bias mitigation.

Kiela et al. (2020) with the release of the Hateful Memes dataset, showed that unimodal architectures—text-only or image-only—consistently perform worse than multimodal ones. Their results provided the fundamental justification for combining textual and visual modalities in meme classification tasks.

Further studies have investigated more complex fusion techniques. Pramanick et al. (2021) presented MOMENTA, a multimodal framework that combines entity-enriched representations with cross-modal attention. The results have shown that attention-based fusion outperforms simple feature concatenation when multiple modalities provide complementary information. Our architecture was established directly on this design principle, using a cross-modal attention layer to allow text representations to attend over CLIP visual features rather than concatenating them.

Regarding the Nepali language, the MemeNePAL dataset was one of the first large collections of Nepali internet memes introduced by Thapa et al. (2025a) with sentiment and hate speech annotations. Multilingual BERT-based text-only models produce comparatively moderate Macro-F1 scores , while multimodal models consistently improve these scores, according to the study. This highlights the challenges of classifying meme sentiment in low-resource languages and the need for multimodal methods.

XLM-RoBERTa (Conneau et al., 2020), which has been pre-trained on 2.5TB of curated Common Crawl text in 100 languages using a SentencePiece BPE vocabulary, has stand out as the top general-purpose multilingual encoder for cross-lingual transfer tests such as XNLI, XQuAD, and MLQA. Its larger variant (with 24 layers, a hidden size of 1024, and 560M parameters) has significantly outperformed the base variant on low-resource languages like Nepali, where the increased model capacity offsets the limited pre-training signal available per language.

In vision-language modeling, CLIP (Radford et al., 2021) has utilized contrastive learning on a vast collection of image-text pairs to create aligned multimodal representations, whereas BLIP-2 (Li et al., 2023) has facilitated effective image caption generation by employing frozen encoders and lightweight transformers. We have used BLIP-2 to generate English-language visual descriptions of each meme, providing the text encoder with a natural-language rendering of the image content

that can be processed in the same token space as the extracted meme text. This design has allowed the model to reason jointly over image content and meme caption without requiring cross-lingual visual grounding.

Prior research further shows that domain-adaptive pre-training improves downstream performance in low-resource settings (Gururangan et al., 2020). Furthermore, methods for generating artificial code-switched text have been proposed to improve cross-lingual feature alignment and model performance on mixed-language tasks. We have incorporated these code-mixed samples into each training fold while deliberately withholding them from validation, ensuring that fold-level metrics remain comparable to the monolingual test distribution(Qin et al., 2020).[1]

3. Data

We have utilized the dataset introduced in prior work (Thapa et al., 2025a,b).The annotation schema follows CrisisHateMM (Bhandari et al., 2023). The dataset has been segmented into training, validation, and test sets containing 1,061, 133, and 133 samples, respectively. It primarily consists of Nepali memes written in Devanagari script, reflecting cultural and political discourse commonly observed in Nepali online communication. The dataset exhibits a class imbalance skewed toward the neutral class, as shown in Table 1.

Sets	Negative	Neutral	Positive	Total
Train	341	473	247	1,061
Validation	39	65	29	133
Test	–	–	–	133
Total	380	538	276	1,194

Table 1: Label Distribution for the Nepali Meme Dataset.

We have additionally incorporated the code-mixed split of the dataset as training augmentation data. This split contains 1,391 Nepali-English code-mixed meme images annotated for sentiment, as shown in Table 2. These samples have been used only during training and have been excluded from all validation splits.

Sets	Negative	Neutral	Positive	Total
Code-Mixed	263	757	371	1,391

Table 2: Label Distribution for the NeMeme Code-Mixed Augmentation Data.

[1] https://zenodo.org/records/15164380

4. Methodology

4.1. Data Preprocessing

4.1.1. Text Preprocessing

Text preprocessing has involved removing URLs, punctuation marks, and special characters to clean the input data. Then the cleaned text has been tokenized using XLM-RoBERTa SentencePiece vocabulary . Since the dataset comprises both monolingual Nepali and code-mixed Nepali–English text in Devanagari script, no transliteration has been applied; the multilingual tokenizer has handled both registers natively.

4.1.2. Image Preprocessing

We have converted all meme images to RGB format. In feature extraction, the images have been resized to 224×224 pixels and normalized to meet the requirements of the CLIP and BLIP-2 models. **Image-Text Similarity Scaling:** Following extracting and normalizing image embeddings,cosine similarity between image and corresponding text embeddings has been calculated. Negative similarity values has been clamped to zero, and the resulting scores were scaled by a factor of 2.5.

4.1.3. OCR Noise Filtering

EasyOCR extraction on code-mixed meme images frequently yields structurally corrupt output due to mixed Devanagari–English script boundaries and stylized meme fonts. Structural analysis of the 1,391 code-mixed OCR samples revealed that 2.9% (40 samples) contained problematic output as high-noise mixed text. Representative examples are shown in Table 3.

Category	Full OCR Output
Noisy	`%"[|=1-[1104; Meurlt "-|11[711`
Noisy	`Relar YE 5 coming Tike IIlrry * :llls :: 1-1!*170)/771!`
Noisy	`ce 9 "T_6 <o[+e dusltr l-ua tluzue u01=8 "=""371-`

Table 3: Examples of noisy OCR output from code-mixed meme images showing full EasyOCR output.

To investigate whether removing these samples improves model robustness, we experimented with three filtering strategies applied exclusively to the code-mixed augmentation data: (i) a Unicode range-based structural filter (Regex), which removes text with fewer than 50% valid Devanagari or Latin characters; (ii) pseudo-perplexity scoring (Salazar et al., 2020) using DistilBERT-Nepali (Sakonii, 2021), a model pre-trained on 13

million Nepali text sequences using masked language modeling, where texts exceeding an auto-calibrated perplexity threshold are removed; and (iii) LLM-based binary classification using Phi-3.5-mini (Abdin et al., 2024), which classifies each OCR text as CLEAN or NOISY. Results are presented in Table 4.

Filter	Scope	OOF F1
Baseline(no filtering)	–	**0.50**
Regex	Codemix only	0.47
DistilBERT-Nepali	Codemix only	0.44
Phi-3.5-mini	Codemix only	0.47

Table 4: Effect of OCR noise filtering strategies on system performance (XLM-R-large + CLIP + BLIP-2). OOF F1 = out-of-fold Macro-F1 on combined train and validation set.

None of the filtering strategies improved OOF F1 over the unfiltered baseline. We attribute this to the dual role of BLIP-2 captions: they provide a language-agnostic visual signal that independently compensates for failed or noisy OCR, while preserving all training samples and their visual representations through CLIP and BLIP-2. Filtering reduces available training data without proportional quality gains, as removed samples retain full visual signal regardless of OCR quality. This finding confirms that the multimodal architecture is inherently robust to OCR noise, rendering explicit filtering unnecessary for this task.

4.2. Data Augmentation

The training set has been augmented at the token level during training to improve generalization. With 25% probability, 15% of input tokens have been replaced with [MASK]; with 20% probability, 10% of words have been randomly dropped; and with 15% probability, adjacent word pairs have been randomly swapped. These stochastic perturbations have been applied only during training and only to the OCR-extracted meme text, not to the image captions. Additionally, 1,391 code-mixed Nepali–English meme samples have been incorporated as training augmentation, with monolingual samples upweighted by a factor of 2 via a WeightedRandomSampler to reduce reliance on the noisier code-mixed source.

4.3. Overview of Experimented Models

4.3.1. Unimodal Models

We have evaluated text-only baselines using three multilingual pre-trained encoders that have been fine-tuned end-to-end on the meme text column: XLM-RoBERTa-base (125M), MuRIL-base-cased (237M), and Twitter-XLM-RoBERTa-base-sentiment (125M). A linear classification head over the [CLS] token has been used. Then the models have been optimized with AdamW using a learning rate of $2e{-}5$, and have been trained for up to 10 epochs with early stopping. The architecture of the unimodal text model has been illustrated in Figure 2.

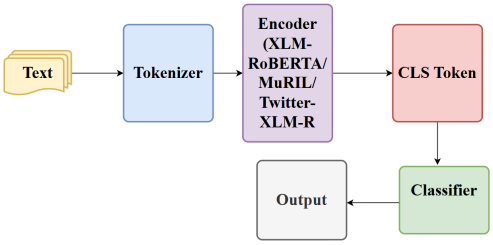


Figure 2: Unimodal Architecture for Text Data Processing and Classification.

4.3.2. Multimodal Models

Our multimodal system has fused three streams: (i) contextual text features from a fine-tuned XLM-RoBERTa-large applied to OCR-extracted meme text combined with BLIP-2 captions (Li et al., 2023); (ii) pooled visual features from a frozen CLIP ViT-L/14 encoder (Radford et al., 2021); and (iii) a BLIP-2 caption embedding along with a CLIP image–text cosine similarity scalar.

Fusion has been performed using a single-layer, 8-head cross-modal attention block with a hidden size of 1024 (128 per head), where the text [CLS] representation from XLM-RoBERTa-large attends over the CLIP image embedding as key and value. The attended text representation has then been added residually and normalized using LayerNorm, and subsequently concatenated with two 256-dimensional projected CLIP streams (image and caption) and a 32-dimensional similarity scalar. The resulting 1568-dimensional vector has passed through a two-layer MLP classifier with GELU activations and dropout ($p = 0.4$). The multimodal architecture has been illustrated in Figure 3.

The XLM-RoBERTa-large backbone (560M parameters) has been partially frozen: the embedding layer and the bottom 18 transformer layers have been frozen, leaving 76.6M trainable parameters. Two learning rates have been used, with a cosine schedule and 10% linear warmup. We have also experimented with NepaliBERT as an alternative text backbone, paired with the same frozen CLIP ViT-L/14 visual stream and fusion head.

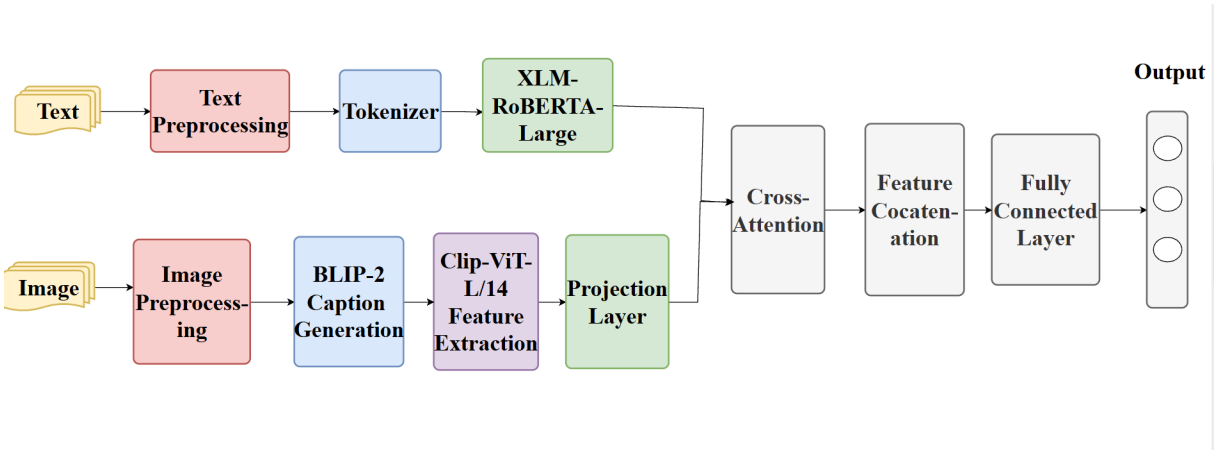


Figure 3: Multimodal Architecture using XLM-RoBERTa-Large and Clip-ViT-L/14.

5. Results and Analysis

This section presents the results of our Nepali meme sentiment classification task, comparing unimodal and multimodal approaches. We have evaluated model performance using macro-averaged precision, recall, and F1 score, with Macro-F1 serving as the primary evaluation metric.

5.1. Comparative Analysis

We have found that among unimodal text classifiers, muril-base-cased performed best, achieving a Macro-F1 of 0.42. It has outperformed xlm-roberta-base (0.40) and twitter-xlm-roberta-base-sentiment (0.39), highlighting the benefit of Nepali-inclusive pretraining.

For multimodal systems, the combination of xlm-roberta-large, CLIP ViT-L/14, and BLIP-2 has achieved the highest overall performance with a Macro-F1 of 0.50, improving +0.08 over the strongest text-only baseline. Specifically, incorporating BLIP-2 captions into the XLM-R-large + CLIP ViT-L/14 system has improved Macro-F1 from 0.42 to 0.50 (+0.08), as shown in Table 6, representing the single largest component-level gain in our ablation. This improvement is attributed to BLIP-2 providing an English-language visual description of each meme, compensating for noisy or empty OCR output in Devanagari script and supplying the text encoder with a language-agnostic rendering of the visual content that complements the raw image embeddings from CLIP.

Replacing cross-attention fusion with simple feature concatenation has reduced the mean OOF Macro-F1 from 0.4775 to 0.4438 (-0.0337), confirming that cross-modal attention is essential for effective modality interaction. Table 5 compares both fusion strategies across all folds.

Ablation studies demonstrate the contribution of each component: replacing CLIP ViT-L/14 with ViT-B/32 has reduced performance to 0.38, substituting XLM-R-large with NepaliBERT has yielded 0.46, and removing BLIP-2 captions has yielded 0.42. These results has confirmed that both aligned visual–textual embeddings and image captions play an important role in multimodal sentiment classification. Table 6 shows the performance of these models.

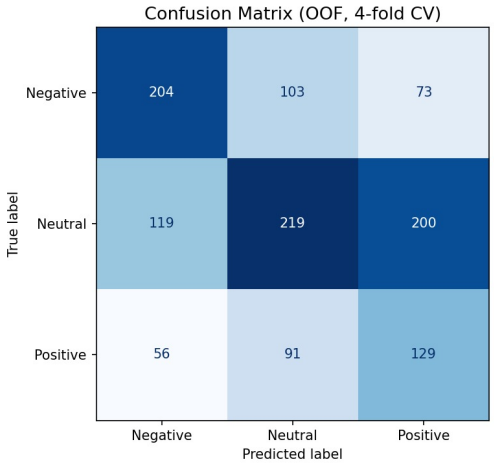


Figure 4: Confusion Matrix of the Multimodal XLM-R-large, CLIP and BLIP-2 model.

Metric	Cross-Attention	Concatenation
Fold 1 Val F1	0.48	0.41
Fold 2 Val F1	0.46	0.46
Fold 3 Val F1	0.51	0.48
Fold 4 Val F1	0.46	0.41
Mean OOF Macro-F1 $\pm$ Std	**0.4775 $\pm$ 0.0205**	**0.4438 $\pm$ 0.0307**

Table 5: Best Macro-F1 Scores for Each Fold and Mean OOF Macro-F1 Comparing Cross-Attention Fusion and Simple Concatenation in 4-Fold Stratified Cross-Validation on the Combined Training and Validation Dataset.

Classifier	Macro Average		
	P	R	F1
Unimodal (Text)			
MuRIL-base	0.44	0.45	0.42
XLM-R-base	0.43	0.41	0.40
Twitter-XLM-R	0.40	0.40	0.39
Multimodal			
XLM-R-base + CLIP ViT-B/32	0.42	0.44	0.38
NepaliBERT + CLIP ViT-L/14	0.46	0.46	0.46
XLM-R-large + CLIP ViT-L/14	0.42	0.44	0.42
XLM-R-large + CLIP + BLIP-2	**0.46**	**0.47**	**0.50**

Table 6: Performance of different systems on the test dataset.

5.2. Error Analysis

To further analyze model performance, we have presented the confusion matrix in Figure 4. It shows that our best model has correctly classified 204 Negative, 219 Neutral, and 129 Positive samples. However, substantial cross-class confusion has remained. For the Negative class (380 total), 103 instances have been misclassified as Neutral and 73 as Positive. Neutral (538 total) has been heavily confused with both extremes, with 119 predicted as Negative and 200 as Positive, which has led to its relatively low recall. Positive (276 total) has been the hardest class overall (F1 = 0.38), with 56 misclassified as Negative and 91 as Neutral, and a notably low precision due to frequent overprediction of Positive (402 total Positive predictions). Overall, the per-class F1 scores have followed the trend: Negative (0.53) > Neutral (0.46) > Positive (0.38). The confusion patterns have suggested strong semantic overlap between Neutral and the polar classes, while Positive has suffered most from class imbalance (23.1% of samples) and boundary ambiguity with Neutral.

Ablation results have further indicated that code-mixed augmentation has degraded performance when OCR quality has been poor. In our Nepali-Code-Mixed dataset, a subset of memes (2.9%) have produced empty or noisy text, which has weakened the text encoder's contribution and has increased misclassification—particularly between Neutral and Positive. The best-performing system has mitigated this issue by (i) incorporating BLIP-2 captions as a language-agnostic visual signal and (ii) upweighting monolingual samples using a WeightedRandomSampler, which has stabilized class learning without excessively biasing predictions toward the minority class. A systematic investigation of explicit OCR noise filtering strategies, detailed in Section 4.1.3, further confirmed that BLIP-2 already provides sufficient noise resilience, as no filtering approach improved over the unfiltered baseline.

5.3. Sensitivity to OCR Quality

The filtering ablation results presented in Section 4.1.3 indicate that the system is highly robust to OCR quality degradation. Structural analysis confirmed that only 2.9% of code-mixed OCR samples were genuinely problematic. The remaining semantic noise — OCR misreadings of colloquial Nepali meme text that appear structurally valid but are linguistically garbled — cannot currently be detected by automated methods without introducing false positives. The consistent failure of all three filtering strategies to improve OOF F1 further confirms that BLIP-2 visual captioning and WeightedRandomSampler together provide sufficient OCR noise resilience, making OCR quality a non-critical factor for this multimodal system.

6. Conclusion

In this research, we have evaluated unimodal and multimodal approaches for assessing the sentiments of Nepali memes. Multimodal systems consistently outperform text-only baselines, with our best system (XLM-RoBERTa-large + CLIP ViT-L/14 + BLIP-2, cross-modal attention fusion) achieving a Macro-F1 of 0.50 on the 4-fold OOF evaluation. A text-only MuRIL-large system with domain-specific warm-up on Nepali tweets achieves 0.4743, demonstrating that task-adaptive pre-training is a strong alternative to multimodal fusion when visual processing pipelines are unreliable. Future work will focus on Nepali-specific OCR, multilingual image captioning, and ensemble combination of the top-performing unimodal and multimodal systems.

7. Limitations

The dataset provided for our task is relatively small and imbalanced, particularly for the positive sentiment classes. Although code-mixed data augmentation has been incorporated, a subset of samples (2.9%) suffer from poor OCR quality, limiting the effectiveness of text features. Explicit OCR noise filtering strategies, including rule-based, transformer-based, and LLM-based approaches, did not improve over the unfiltered baseline, suggesting that more sophisticated noise detection methods tailored to colloquial low-resource meme text remain an open challenge. Furthermore the frozen visual backbone and limited text augmentation may not fully capture subtle multimodal cues, leading to some misclassifications.Additionally, our system has not employed end-to-end vision-language models (VLMs) such as LLaVA or InstructBLIP, which jointly reason over image and text in a unified framework and may better capture the nuanced visual-linguistic interactions present in meme con-

tent. Lastly, the performance of the model could be improved with more diverse and high-quality examples, especially for underrepresented memes.

8. Ethics Statement

In this study, we have developed our methodology following the highest ethical practices. By contributing to the identification of hate and sentiment understanding in low-resource memes, we hope to make the internet a safer and more inclusive place. We are committed to sharing our findings to prevent offensive and hateful content online while respecting linguistic and cultural diversity.

9. References

Marah Abdin et al. 2024. Phi-3 technical report. https://arxiv.org/abs/2404.14219.

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8342–8360.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Thanh Tin Nguyen, Nhat Truong Pham, Ngoc Duy Nguyen, Hai Nguyen, Long H Nguyen, and Yong-Guk Kim. 2022. Hcilab at memotion 2.0 2022: Analysis of sentiment, emotion and intensity of emotion classes from meme images using single and multi modalities (short paper). In *DE-FACTIFY@ AAAI*.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. Momenta: A multimodal framework for detecting harmful memes and their targets. *arXiv preprint arXiv:2109.05184*.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp. *arXiv preprint arXiv:2006.06402*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Sakonii. 2021. distilbert-base-nepali. https://huggingface.co/Sakonii/distilbert-base-nepali.

Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of ACL*.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Limor Shifman. 2013. *Memes in digital culture*. MIT press.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and

sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

MEME-Fusion@CHiPSAL 2026: Multimodal Ablation Study of Hate Detection and Sentiment Analysis on Nepali Memes

Samir Wagle*, **Reewaj Khanal***, **and Abiral Adhikari***
Department of Computer Science and Engineering, Kathmandu University, Nepal
{waglesameer5, emailreewaj, abiraladhikari1222}@gmail.com
* These authors have contributed equally.

Abstract

Hate speech detection in Devanagari-scripted social media memes presents compounded challenges: multimodal content structure, script-specific linguistic complexity, and extreme data scarcity in low-resource settings. This paper presents our system for the CHiPSAL 2026 shared task, addressing both Subtask A (binary hate speech detection) and Subtask B (three-class sentiment classification: positive, neutral, negative). We propose a hybrid cross-modal attention fusion architecture that combines CLIP (ViT-B/32) for visual encoding with BGE-M3 for multilingual text representation, connected through 4-head self-attention and a learnable gating network that dynamically weights modality contributions on a per-sample basis. Systematic evaluation across eight model configurations demonstrates that explicit cross-modal reasoning achieves a 5.9% F1-macro improvement over text-only baselines on Subtask A, while uncovering two unexpected but critical findings: English-centric vision models exhibit near-random performance on Devanagari script, and standard ensemble methods catastrophically degrade under data scarcity ($N \approx 850$ per fold) due to correlated overfitting. The code can be accessed at https://github.com/Tri-Yantra-Technologies/MEME-Fusion/

Keywords: Multimodal Fusion, Vision-Language Models (VLMs), Hate Speech and Sentiment Detection, Multimodal Hate Speech Detection, Devanagari Script, Low-Resource Languages, Meme Analysis, Social Media Moderation.

1. Introduction

The proliferation of hate speech on social media platforms has become a critical concern in the moderation of online content, with particular severity in multilingual and multimodal contexts. While English text-based hate speech has received substantial research attention, non-English languages employing non-Latin scripts such as Devanagari remain significantly underserved. This gap is consequential: Hindi, Nepali, Marathi, and related Devanagari-scripted languages collectively serve over 600 million speakers who increasingly engage with social media through meme formats, a modality that fuses visual and textual signals into a semantically unified artifact. The CHiPSAL 2026 shared task Thapa et al. (2026), Sarveswaran et al. (2026) on Multimodal Hate and Sentiment Understanding in Low-Resource Text-Embedded Images directly addresses this challenge. Our system participates in two subtasks: Subtask A (binary hate speech detection: hate vs. non-hate) and Subtask B (three-class sentiment classification: positive, neutral, negative). Both subtasks operate on a curated corpus of 1,068 Devanagari meme instances, requiring joint reasoning over OCR-extracted text and meme image content. Three structural challenges characterize this problem. First, the dataset is severely limited in scale (N = 1,068 total), creating an acute sample-to-parameter scarcity that destabilizes high-dimensional fusion strategies. Second, class imbalance is pronounced in Subtask A, with hate instances constituting 67.4% of the corpus. Third, standard pretrained vision models such as CLIP were predominantly trained on English web data, fundamentally limiting their representational capacity for Devanagari script and culturally specific Nepali visual content. To address these challenges, we propose a hybrid cross-modal attention fusion architecture. At its core, 4-head self-attention computes explicit dependencies between visual embeddings from CLIP and textual embeddings from BGE-M3, followed by a learnable softmax-normalized gating network that assigns per-sample modality weights. This allows the model to privilege text when linguistic hate signals are explicit (e.g., direct slurs) and to upweight visual features when textual content is ambiguous. Critically, our ablation studies reveal that naive approaches: simple concatenation and traditional ensemble methods actively harm performance in this low-resource regime, a finding with direct practical consequences for NLP system design in data-scarce settings.

2. Related Work

The research trajectory for computational hate speech detection within Devanagari-scripted languages has been predominantly catalyzed by the Hate Speech and Offensive Content (HASOC) shared task series, which served to establish crucial benchmark datasets and standardized evaluation protocols, primarily for Hindi Mandl et al. (2019).

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 300–307
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

The methodological approaches ranging from classical feature engineering, such as Term Frequency-Inverse Document Frequency (TF-IDF) and character n-grams, to sophisticated transformer-based architectures, such as multilingual BERT variants are continuously evolving. Although these deep learning models consistently demonstrate superior performance over conventional baselines, these models are less effective when computational resources are limited. When fine tuning data is scarce, the model tend to provide English-centric biases that can be mitigated by more robust datasets. This evolving architectural landscape is further complemented by earlier deep learning efforts, notably the Bi-directional Long Short-Term Memory (Bi-LSTM) framework Adhikari et al. (2024), which successfully delineated between profanity and general offensiveness in Nepali and recorded a high-watermark accuracy of 87.8% on an analogously data-constrained corpus. More cutting-edge investigations are now focused on leveraging Large Language Model (LLM)-based paradigms, employing zero-shot and few-shot learning for South Asian content moderation Thapa et al. (2025a), as well as exploring retrieval-augmented distillation strategies to strategically transfer and exploit knowledge representations acquired from data-rich languages Thapa et al. (2025b).

The extension from text-only to multimodal hate detection reflects the growing predominance of meme-based communication on social media. Bhandari et al. (2023) introduced the CrisisHateMM dataset focusing on hate speech in text-embedded images from geopolitical conflict contexts, demonstrating that joint vision-language models outperform unimodal baselines by 8–12% F1. Thapa et al. (2025a) further explored prompt-based frameworks for code-mixed and low-resource meme understanding, showing that carefully engineered prompt templates improve model performance even under data scarcity. These works primarily evaluate fusion strategies on datasets with thousands of labeled instances; their applicability to extreme low-resource regimes ($N < 1{,}100$) remains an open question addressed by our study.

OpenAI's CLIP model Radford et al. (2021), pretrained on 400 million image-text pairs from the English web, provides strong zero-shot generalization for Latin-script and Western visual content. On top of this multilingual vision-language pretraining(mCLIP, IndicCLIP) was carried out to prevent performance degradation on non-Latin scripts. However, Devanagari coverage in these models still remains limited. For text encoding, the BAAI General Embedding model BGE-M3 Xiao et al. (2023) offers strong multilingual dense retrieval performance with explicit support for Indic scripts, outperforming mBERT and XLM-RoBERTa on low-resource language retrieval benchmarks. The complementarity of CLIP's visual capabilities and BGE-M3's multilingual text understanding motivates our dual-encoder architecture.

3. Dataset, Tasks, and Preprocessing

3.1. Task Definition

The CHiPSAL 2026 shared task provides a corpus of Devanagari meme instances, each consisting of a JPEG image file containing visual content with embedded Devanagari text, and a text field containing OCR-extracted Devanagari/Nepali text. Our system addresses two subtasks:

- Subtask A – Binary Hate Speech Detection: Given a meme, classify it as Hate (1) or Non-Hate (0).
- Subtask B – Sentiment Classification: Given a meme, classify it as Positive (2), Neutral (1), or Negative (0).

3.2. Data Distribution and Class Imbalance

The dataset comprises 1,068 total meme instances evaluated via stratified 5-fold cross-validation (random seed = 42). Table 1 presents the class distribution for Subtask A:

Class	Label	Training	Evaluation
Non-Hate	0	50	50
Hate	1	50	35
Total		100	85

Table 1: Class distribution for Subtask A (Binary Hate Speech Detection)

Similarly, the class distribution for Subtask B is shown in Table 2:

Class	Label	Training	Evaluation
Positive	2	50	29
Neutral	1	50	50
Negative	0	50	39
Total		150	118

Table 2: Class distribution for Subtask B (Multi-class Sentiment Detection)

The corpus exhibits a 2.07:1 Hate-to-Non-Hate imbalance, posing a systematic risk of majority-class exploitation during training. For Subtask B, the sentiment distribution is approximately 45%

Negative, 32% Neutral, and 23% Positive, presenting a three-class variant of the same challenge. Macro-averaged F1 score is adopted as the primary evaluation metric for both subtasks, as it penalizes models that over-predict majority classes.

3.3. Preprocessing Pipeline

3.3.1. Text Preprocessing

OCR-extracted Devanagari/Nepali text was tokenized using the BAAI/bge-m3 subword segmenter (SentencePiece). Sequences were padded or truncated to a maximum length of 77 tokens, consistent with CLIP's context window constraint. We deliberately preserved all raw social media elements:hashtags, emojis, URL fragments, as these may carry contextually relevant signals.

3.3.2. Image Preprocessing

Images were processed through the CLIP processor pipeline: RGB conversion, resizing (shorter edge to 224 pixels), center cropping to 224×224 pixels, and normalization using ImageNet channel statistics (mean: [0.485, 0.456, 0.406]; std: [0.229, 0.224, 0.225]).

3.3.3. Text-Removed Image Variant

To isolate the contribution of visual layout from in-image typography, we generated a complementary image dataset using a Tesseract OCR detection pipeline. Then, detected Devanagari text regions were replaced via Gaussian blur inpainting , producing images that preserve background, color, and spatial composition while eliminating typographic content.

4. Methodologies

For this experiment we prepared an ablation study setup using different encoders both image and text, feature selection and fusion strategies to observe the effect of different factors on performance of models.

4.1. Pretrained Encoders

4.1.1. Visual Encoder: CLIP ViT-B/32.

CLIP encodes each 224×224 image in a 512-dimensional dense embedding via a 12-layer Vision Transformer. To balance task-specific adaptation with preservation of generic visual representations, we freeze the lower 10 layers and fine-tune only the final 2 transformer layers. This selective fine-tuning strategy is critical given the limited size of the training set.

4.1.2. Language Encoder: BGE-M3.

BGE-M3 exhibited superior performance during initial experiments conducted to evaluate multilingual capabilities using BGE-M3, E5-large, and XLM-RoBERTa. With the purpose of conserving limited computational resources during the extensive 8-model ablation study, BGE-M3 was selected as the base text embedding model for all subsequent experiments. BGE-M3 generates mean-pooled, L2-normalized embeddings of dimensionality 1024 from its 24-layer transformer. We fine-tune the final 4 layers (layers 21–24) to adapt the model's semantic representations to the hate speech domain while preserving the core multilingual embedding space established during pretraining. The selection of BGE-M3 is empirically supported by recent extensive evaluations on Indic language benchmarks, where it has demonstrated superior recall and semantic representation capabilities for low-resource and code-mixed scripts compared to traditional multilingual models Doddapaneni et al. (2023)

4.2. Model Configurations

We systematically evaluated eight model configurations across three structural paradigms: unimodal baselines, conventional fusion methodologies, and our proposed hybrid attention-based architecture. All configurations employ a standardized multi-layer perceptron classifier to ensure comparative rigor. The unimodal baselines isolate discrete modalities, with M1 operating exclusively on BGE-M3 textual embeddings, while M2 and M3 process visual features extracted from original and text-ablated images, respectively. Among the standard fusion approaches, M4 implements early fusion via direct feature concatenation, whereas M5 and M6 utilize late fusion strategies, specifically soft voting and bagging ensembles, to aggregate independent predictive probabilities. The necessity of our proposed hybrid approach is echoed by recent findings in multimodal hate speech detection, which demonstrate that dynamic cross-attention mechanisms explicitly capturing semantic alignment between text and visual cues significantly outperform naive concatenation or late voting strategies Kitukale et al. (2025). Finally, M7 and M8 instantiate our novel dynamic hybrid fusion, applied to the original and text-ablated image datasets. The proposed hybrid framework (M7-M8) executes a dynamic, sample-specific fusion mechanism engineered to optimally integrate cross-modal signals. Initially, both textual and visual embeddings are projected into a shared, lower-dimensional latent space; this dimensionality reduction prevents the higher-dimensional text representations from mathematically dominating the visual features. These harmonized embeddings are subsequently processed through a multi-head self-

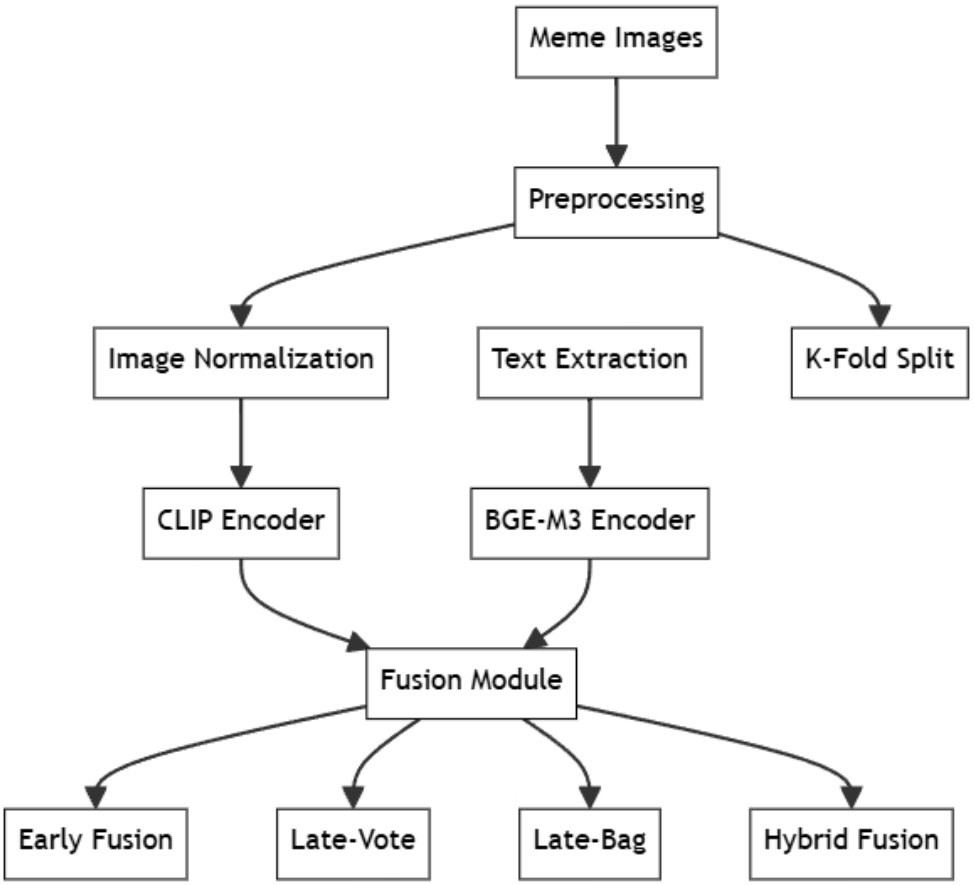


Figure 1: System Architecture Diagram

attention layer, enabling explicit interaction and mutual conditioning between the visual and linguistic modalities. Ultimately, a learnable gating network operates at the instance level to dynamically assign modality-specific weights. This configuration empowers the architecture to adaptively prioritize either explicit linguistic markers or visual context, contingent upon which modality exhibits the highest discriminative utility for a given input.

4.3. Training Configuration

All models are trained using AdamW optimization with decoupled weight decay. To mitigate the 2.07:1 class imbalance, we employ frequency-weighted cross-entropy loss with a suppression exponent = 0.3, yielding class weights W0 = 1.258 (Non-Hate) and W1 = 1.000 (Hate). Label smoothing coefficient of 0.1 prevents overconfident predictions. The optimized fusion models (M4-M8) use a learning rate of 2e-5, weight decay of 1e-2, dropout of 0.5, and ReduceLROnPlateau scheduling with factor of 0.5 and patience 5 epochs. Early stopping with patience 10 prevents overfitting. All models are evaluated via stratified 5-fold cross-validation to ensure robust performance estimates given the limited dataset size.

5. Results and Evaluation

The practical efficacy of our proposed architecture was evaluated in an un-seen test set of CHIPSAL benchmark of meme dataset, via evaluation of our submission of predictions in codabench [1]. The official benchmark leaderboard ranking of our models for Subtask A (Hate Speech Detection) and Subtask B (Sentiment), were 4th and 5th respectively. The elaborated findings of our experimentations are further discussed below.

5.1. Subtask A: Binary Hate Speech Detection

Table 3 presents cross-validated performance for all eight model configurations on Subtask A. The hy-

[1] https://www.codabench.org/competitions/12090/ https://www.codabench.org/competitions/12091/

brid fusion model (M7) achieves the highest macro-F1 of 0.6834, outperforming the text-only baseline by 5.9% absolute and exceeding early concatenation fusion by 11.3%.

The performance hierarchy (Hybrid Fusion > Late Voting > Text-Only > Early Fusion > Bagging > Image-Only) suggests that the strategy for modality interaction is the primary driver of performance variation. Notably, the 16.6% gap between text-only (0.6455) and image-only (0.5535) models establishes the primacy of the linguistic modality in Devanagari hate meme classification.

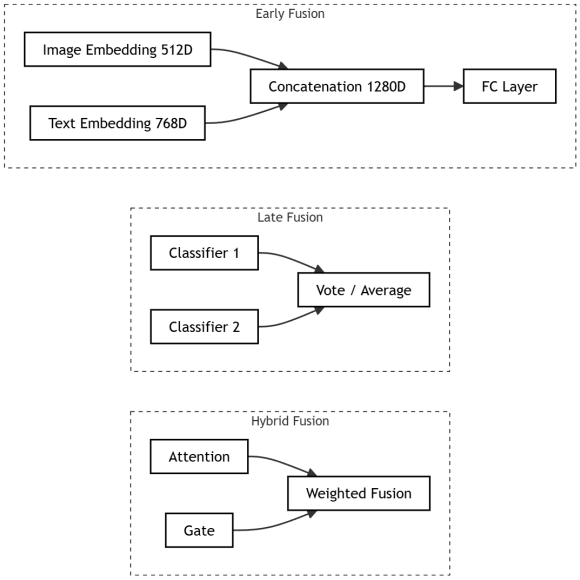


Figure 2: Comparison of multi-modal fusion strategies

5.2. Subtask B: Sentiment Classification

Table 4 reports best results for Subtask B (three-class sentiment: Positive, Neutral, Negative) . The three-class formulation amplifies the difficulty of the task, reducing all models' F1 scores relative to the binary Subtask A. Nevertheless, the same architectural hierarchy holds: the hybrid fusion model (M7) achieves the highest macro-F1 of 0.6102, improving over the text-only baseline by 2.2% and over early concatenation by 7.9%.

The Neutral class presents the greatest classification difficulty across all models, with per-class F1 scores consistently 8-12 points below Positive and Negative classes. This reflects the semantic ambiguity of neutral content, which lacks the distinctive lexical markers characteristic of hate (Subtask A) or affective polarity (sentiment extremes).

5.3. Class-Specific Performance: The Non-Hate Detection Gap

Table 5 decomposes Subtask A performance by class for the two strongest models, revealing a critical behavioral difference between M1 and M7. M7 improves Non-Hate recall by 14 percentage points (0.73 vs. 0.59), creating a substantially more balanced classification profile. This improved minority-class detection is precisely what macro-F1 scoring rewards, explaining why M7 ranks first on the primary metric despite its lower accuracy (0.7015 vs. 0.7239 for M1). This is not a secondary concern: in deployed content moderation systems, systematically failing to identify non-hateful content constitutes an over-censorship failure mode with direct impact on user experience and free expression.

6. Discussion

The superior performance of M7 (Hybrid Fusion) arises from its use of structured cross-modal conditionality rather than simple feature aggregation. Unlike early concatenation methods, M7's self-attention mechanism enables mutual representation grounding. This is reflected in the high attention weights for text-to-image (0.73) and image-to-text (0.68) tasks, which facilitate entity localization and visual disambiguation. Furthermore, M7's dynamic gating provides a level of per-sample adaptivity that late-voting (M5) lacks; by routing 68% of instances as text-dominant based on signal quality, the model successfully overcomes rigid architectural priors.

A key systemic finding is that image-only models (M3) achieve near-random performance on Devanagari content, with a macro-F1 of only 0.5535. Because CLIP's pretraining distribution is heavily English-centric, it struggles to interpret Indic semantics or culturally specific Nepali iconography. Rather than understanding the content, the model relies on weak visual heuristics, such as text density ($\rho = 0.31$) and typographic weight ($\rho = 0.29$). This suggests that deploying standard Vision-Language Models (VLMs) for South Asian content moderation is structurally unreliable without dedicated multilingual adaptation.

Counterintuitively, standard ensemble methods collapsed under the pressure of extreme data scarcity. Bagging (M6) underperformed simple soft voting (M5) by 3.8% due to correlated overfitting. With a small sample size ($N = 854$), bootstrap resampling produced highly overlapping folds ($\rho = 0.87$), which caused the ensemble to amplify noise rather than cancel it out. Similarly, early concatenation (M4) suffered from the curse of dimensionality; its low sample-to-dimension ratio (0.556) forced the MLP to learn high-variance patterns rather than generalizable features. In such low-

Rank	Model Configuration	F1 Macro	Accuracy	Precision	Recall
1	M7: Hybrid Fusion (Cross-Attn + Gating)	0.6834	0.7015	0.6817	0.7023
2	M5: Late Fusion (Soft Voting)	0.6511	0.6940	0.6521	0.6503
3	M1: Text-Only Baseline (BGE-M3)	0.6455	0.7239	0.6886	0.6376
4	M8: Hybrid Fusion (Text-Removed Images)	0.6436	0.7015	0.6553	0.6384
5	M6: Late Fusion (Bagging, k=3)	0.6267	0.6567	0.6245	0.6341
6	M4: Early Fusion (Concatenation)	0.6140	0.6940	0.6425	0.6096
7	M2: Image-Only (Original Images)	0.5952	0.6791	0.6205	0.5927
8	M3: Image-Only (Text-Removed)	0.5535	0.6343	0.5636	0.5535

Table 3: Cross-validated performance on Subtask A (Binary Hate Speech Detection). Highlighted row indicates primary system.

Rank	Model Configuration	F1 Macro	Accuracy	Precision	Recall
1	M7: Hybrid Fusion (Cross-Attn + Gating)	0.6102	0.6714	0.6095	0.6231
2	M1: Text-Only Baseline (BGE-M3)	0.5881	0.6890	0.6014	0.5763
3	M5: Late Fusion (Soft Voting)	0.5744	0.6621	0.5791	0.5702
4	M4: Early Fusion (Concatenation)	0.5312	0.6409	0.5498	0.5181
5	M2: Image-Only (Original Images)	0.4893	0.5831	0.5012	0.4821

Table 4: Cross-validated performance on Subtask B (Sentiment Classification: Positive/Neutral/Negative). Selected configurations shown.

Model	Class	Precision	Recall	F1
M1: Text-Only	Non-Hate (0)	0.68	0.59	0.63
M1: Text-Only	Hate (1)	0.70	0.69	0.69
M7: Hybrid Fusion	Non-Hate (0)	0.64	0.73	0.68
M7: Hybrid Fusion	Hate (1)	0.72	0.67	0.69

Table 5: Per-class precision, recall, and F1 for M1 (Text-Only) and M7 (Hybrid Fusion) on Subtask A.

resource regimes, the dimensionally constrained fusion used in M7 becomes a mathematical necessity.

Our evaluation reveals a critical 'Accuracy-F1 paradox' within the baseline models. The text-only model (M1) achieved a peak Subtask A accuracy of 0.7239, but it did so by aggressively exploiting a 2.07:1 class imbalance. By over-predicting the majority 'Hate' class, M1 sacrificed minority-class recall a trend also observed in prior work on Nepali profanity detection Adhikari et al. (2024). While these unimodal frameworks excel at spotting overt lexical slurs, they fail to grasp the implicit visual context required for modern multimodal hate speech. M7 resolves this tension, yielding the highest macro-F1 (0.6834) and closing a significant 14-point gap in 'Non-Hate' recall (0.73 vs. 0.59). These results prove that optimizing for raw accuracy in imbalanced moderation tasks merely masks systemic failures and risks the widespread over-censorship of benign content.

An error analysis of M7 isolates two primary failure boundaries that currently exceed encoder capacities. First, false positives (52%) are largely driven by a 'pragmatic deficit' regarding political satire; while the model detects visual hostility, it consistently fails to infer comedic or sarcastic intent. Second, false negatives (56%) are often the result of 'cultural masking.' In these instances, implicit political threats and code-mixed Out-Of-Vocabulary (OOV) tokens effectively hide the hateful intent, suppressing semantic signals before they can be processed by the transformer layers. These failures highlight the need for models that understand not just the language, but the specific cultural and social shorthand of the South Asian digital landscape.

7. Conclusion

This paper presents a hybrid cross-modal attention fusion system for multimodal hate speech detection

and sentiment classification in Devanagari-scripted memes, evaluated on the CHiPSAL 2026 shared task. Our primary system achieves a macro-F1 of 0.6834 on Subtask A and 0.6102 on Subtask B, demonstrating that dynamic cross-modal reasoning through attention and learnable gating can yield meaningful improvements over both unimodal baselines and naive fusion methods in low-resource settings. Beyond the primary performance results, the study yields three findings of broader significance. First, CLIP-based vision models are fundamentally limited for Devanagari content, achieving near-random performance and exposing a critical gap in current vision-language pretraining coverage of South Asian scripts. Second, ensemble and naive fusion methods violate their theoretical assumptions under extreme data scarcity, actively degrading performance compared to simpler constrained approaches. Third, macro-F1 and class-specific recall provide substantially more informative evaluation signals than overall accuracy in imbalanced hate speech tasks, and their adoption should be considered standard practice in this domain.

7.1. Limitations

The primary limitation of this study is dataset scale. With only 854 training samples per fold, aggressive regularization was required throughout, constraining model capacity and limiting the depth of fine-tuning possible without overfitting. The preprocessing pipeline's reliance on Tesseract OCR for Devanagari text extraction introduces character-level errors (approximately 22% error rate on social media content) that propagate through the text encoder. The corpus's tight temporal and regional scope-Nepali social media from 2022-2023, predominantly drawn from a single national election cycle-limits generalization to other Devanagari-speaking regions (e.g., Hindi-speaking India, Marathi-speaking Maharashtra) and to content produced after this period.

7.2. Future Work

Four directions are prioritized for future work. First, data augmentation strategies-including back-translation, synthetic meme generation via multimodal LLMs, and cross-lingual transfer from Hindi hate speech corpora-could substantially expand effective training set size. Second, the integration of natively multilingual vision-language models with Indic script pretraining coverage (e.g., IndicCLIP) would directly address CLIP's Devanagari blindness. Third, incorporating cultural knowledge bases and context-aware pragmatic reasoning modules could help bridge the gap exposed by coded language failure modes. Finally, extension of the hybrid attention architecture to Subtask B's three-class sentiment formulation with class-specific gating strategies may improve performance on the semantically ambiguous neutral class.

Ethical Considerations

Automated hate speech detection systems carry inherent risks of false positives that could unjustly suppress legitimate political satire, critical commentary, or minority-group discourse. Our error analysis demonstrates that 52% of M7's false positives arise precisely from satirical content, underscoring the need for human-in-the-loop review for borderline predictions in deployed systems. Furthermore, because hate speech definitions are culturally and contextually contingent, models trained in this localized Nepali corpus should not be generalized to other South Asian communities without domain-specific retraining and consultation with community stakeholders. We acknowledge the emotional labor of annotators who labeled potentially traumatic content and advocate for appropriate psychological support and fair compensation in future annotation efforts.

Acknowledgements

We thank the CHiPSAL 2026 organizers for curating the dataset and coordinating the shared task, and the anonymous reviewers for their constructive feedback.

References

Abiral Adhikari, Prashant Manandhar, Reewaj Khanal, Samir Wagle, Praveen Acharya, and Bal Krishna Bal. 2024. Profanity and offensiveness detection in Nepali language using bi-directional LSTM models. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 515–521, AU-KBC Research Centre, Chennai, India. NLP Association of India (NLPAI).

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra,

Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages.

Gauri Kitukale, Navneet Pratap Singh, Sidharth Quamara, et al. 2025. Ddcaf: Dynamic dual cross-attention fusion framework for multimodal hate speech detection. *Research Square*. Preprint, Version 1.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohsen Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 14–17.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025a. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025b. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

eGrantha.ai@CHiPSAL 2026: Stochastic Image Captioning for Robust Hate Speech Detection in Low-Resource Nepali Memes

Anish Thapaliya
Institute of Science and Technology, Tribhuvan University
Kathmandu, Nepal
anish.805522@sms.tu.edu.np

Abstract

This paper presents a system for hate speech detection in low-resource Nepali memes, submitted as part of Subtask A of the Shared Task on Multimodal Understanding at CHiPSAL 2026. Detecting hateful memes is particularly challenging due to the combination of images, text, and emojis used to portray humor, satire, or sociopolitical commentary, as well as the low-resource nature of the Nepali language. We investigate a range of unimodal and multimodal modeling strategies, including text-only, vision-text, and caption-based approaches. For caption generation, the Gemini family of models (Gemini 2.X and Gemini 3.X) was used to produce contextually rich captions, which are publicly released as **NeMeme-CAP** on Hugging Face. Caption-based modeling leverages stochastic caption augmentation to address class imbalance and Test-Time Augmentation (TTA) to reduce prediction variance and improve model robustness. The best-performing system fine-tunes an encoder-only transformer model, RoBERTa-base, on the generated captions, achieving **third** place on the official leaderboard with a macro-averaged F1-score of **0.7397**. The code is publicly available at https://github.com/thapaliya123/LREC-CHiPSAL-2026.

Keywords: multimodal hate speech detection, Nepali memes, low-resource Nepali NLP, vision-language model, stochastic caption augmentation

1. Introduction

Social media platforms such as Facebook, Instagram, Twitter (currently X) have popularized memes as a form of multimodal communication that combines images, text, and emojis to portray humor, satire, or sociopolitical commentary (as shown in Figure 1). Despite the popularity of memes, their multimodal nature facilitates the proliferation of hate speech targeting individuals and communities on the basis of race, gender, religion, or nationality (Kiela et al., 2020). Online hate speech has become a growing concern, prompting increasing research efforts toward automated detection methods (Parihar et al., 2021). However, since it is challenging to manually keep up with everything being shared, prior research and shared tasks have primarily focused on developing automated systems for high-resource languages such as English (Thapa et al., 2025a); comparatively limited attention has been given to low-resource languages such as Nepali, a Devanagari-script language spoken by more than 20 million people worldwide.

To address this gap, a shared task (Thapa et al., 2026) was organized as part of the CHiPSAL 2026 workshop (Sarveswaran et al., 2026). Subtask A was framed as a binary classification problem, where each meme was labeled as either hate or non-hate. The official leaderboard ranking was based on macro-averaged F1-score as the primary metric.

To address this task, several transformer-based strategies were explored. The best-performing solution uses the **Gemini model variants** (Comanici et al., 2025; Google DeepMind, 2025b,a) to generate context-aware image captions, leveraging their strong image understanding capabilities. The RoBERTa-base (Liu et al., 2019) model was fine-tuned on the generated captions for binary classification (non-hate or hate). Additionally, Test-Time Augmentation (TTA) (Moshkov et al., 2020) was applied, which further improved the system performance.



Figure 1: Examples of meme data.

The main contributions are as follows:

- A systematic comparison of pre-trained transformer encoders fine-tuned on Gemini-generated contextually rich captions for meme hate speech detection.
- A demonstration of the efficacy of a stochastic caption regeneration strategy for class-imbalance mitigation, where TTA via caption diversity further enhanced the test F1-score at no additional training cost.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 308–315
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

Class	Train	Validation	Test
non-hate	348	35	-
hate	720	98	-
Total	**1068**	**133**	**134**

Table 1: Class-wise distribution of samples for Subtask A.

- **NeMeme-CAP** [1], a publicly released dataset of Gemini-generated captions for Nepali memes.
- The system ranked **third** on the official CHiPSAL 2026 leaderboard with a macro F1-score of **0.7397**.

,

2. Dataset and Task

The dataset from Subtask A of the Shared Task (Thapa et al., 2026) was utilized, where each meme was categorized as either hate or non-hate. The dataset was compiled from prior works, including NeMeme (Thapa et al., 2025b), ENeMeme (Thapa et al., 2025c), and CrisisHateMM (Bhandari et al., 2023).

The class-wise distribution of training, validation, and test splits is reported in Table 1.

3. Methodology

The system includes a three-stage pipeline: **(i)** image caption generation; **(ii)** stochastic caption augmentation; and **(iii)** transformer fine-tuning.

Formally, given a meme image I and a structured prompt P that instructs the model to describe visible content, the vision-language model (VLM), acting as a caption generator G, produces context-aware image captions in natural language:

$$\hat{y} = C\big(G(I,\, P)\big) \tag{1}$$

where C denotes the downstream text classifier and $\hat{y}$ the predicted hate-speech label. The caption $T = G(I, P)$ is the only textual input to the text classifier, converting the multimodal problem into a standard text classification problem and eliminating the need for a separate OCR pipeline, cross-modal fine-tuning, or Nepali-specific vision–language pretraining.

Figure 2 illustrates the overall training and inference pipeline of the experiment.

3.1. Image Caption Generation

The **Gemini 3 Flash** (Comanici et al., 2025; Google DeepMind, 2025a) was accessed via the Google Generative AI API using a structured prompt to generate context-aware image captions, which are publicly released as **NeMeme-CAP** on Hugging Face. The full structured prompt is provided in Appendix A.

[1] https://huggingface.co/datasets/Anish/nepali-meme-captions

Class	Before Aug.	After Aug.
non-hate	348	696
hate	720	720
Total	**1068**	**1416**

Table 2: Class-wise distributions of the training set before and after applying stochastic caption augmentation (Aug.).

3.2. Stochastic Caption Augmentation

To mitigate class imbalance observed in the training set (see Table 1), a data augmentation strategy leveraging the stochastic nature of VLM decoding was explored, in which Gemini 3 Flash was requeried at **temperature 1.0**, producing diverse descriptive captions of the same meme image. Concretely, non-hate class training images received one additional caption, doubling minority-class coverage and yielding a more balanced training distribution (see Table 2).

At inference time, the same stochastic decoding mechanism was applied as **TTA**. A diverse family of Gemini model variants – **Gemini 2.5 Flash**, **Gemini 2.5 Pro** (Comanici et al., 2025), **Gemini 3 Flash** (Google DeepMind, 2025a), and **Gemini 3 Pro** (Google DeepMind, 2025b) – generated five independent captions for each test image, and each sample was passed through the trained classifier separately. The resulting class-probability distributions were averaged before argmax was taken to produce the final prediction (see Figure 2).

3.3. Transformer Fine-Tuning

Caption classification was framed as standard sequence classification employing transformer architectures (Vaswani et al., 2017). The generated caption T was tokenized and encoded by a pretrained transformer encoder, yielding a contextualized `[CLS]` token representation:

$$\mathbf{h} = \mathrm{Encoder}(T) \in \mathbb{R}^d, \tag{2}$$

where d denotes the hidden dimension of the encoder.

The `[CLS]` token representation was projected to a binary hate-speech prediction through a linear classification head:

$$\hat{y} = \mathrm{softmax}(\mathbf{W}\mathbf{h} + \mathbf{b}), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{2\times d}$ is the learned projection matrix and $\mathbf{b} \in \mathbb{R}^2$ is the bias term.

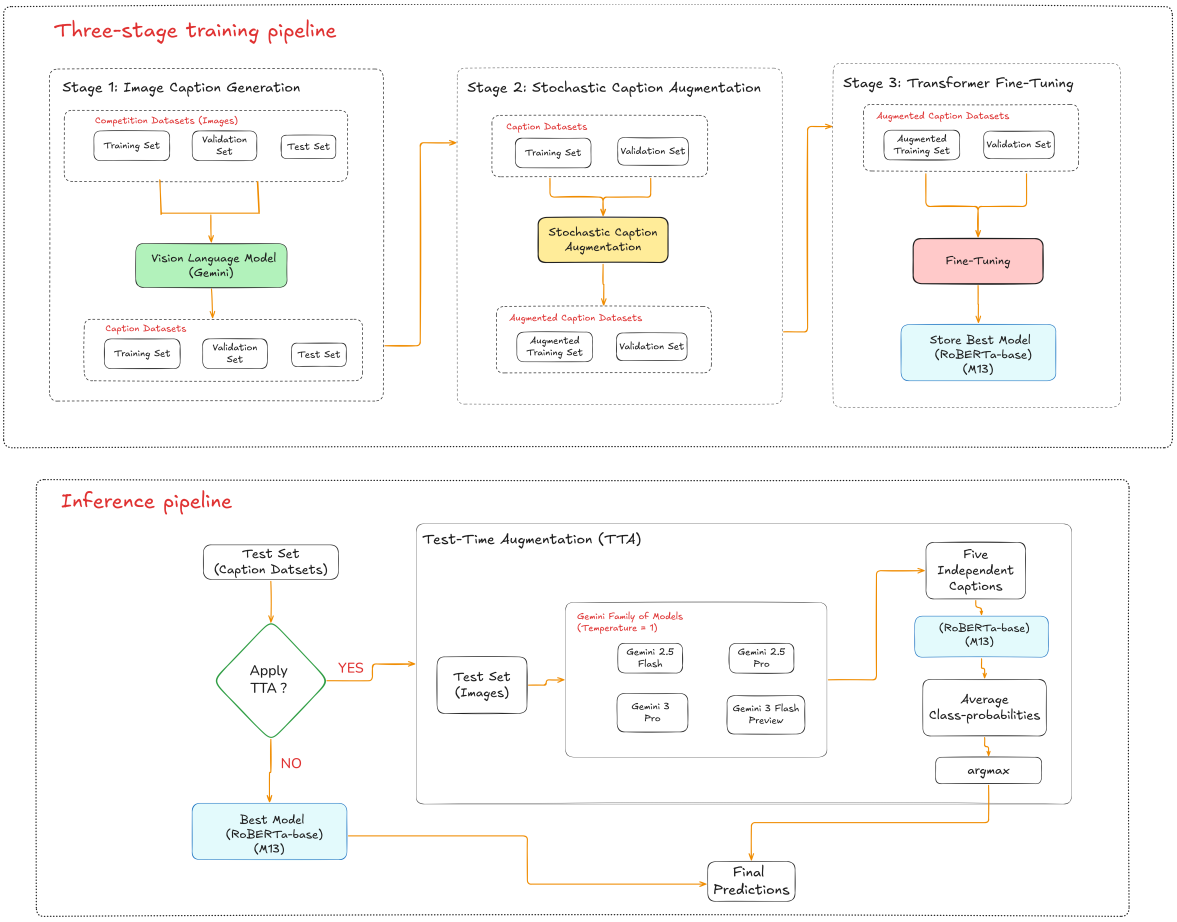


Figure 2: Three-stage training (top) and inference (bottom) pipelines. Stage 1 converts meme images into natural language captions using Gemini 3 Flash (Google DeepMind, 2025a). Stage 2 applies stochastic caption augmentation with a temperature hyperparameter set to 1.0 to address class imbalance. Stage 3 fine-tunes a pre-trained transformer encoder on the augmented caption dataset and stores the checkpoint with the highest validation macro F1. During inference, a total of five independent captions per test image are generated using a family of Gemini model variants (Comanici et al., 2025; Google DeepMind, 2025b) at temperature 1. The resulting class probability distributions are averaged, and the final prediction is obtained via the argmax operation.

The following encoder-only transformer architectures were evaluated, spanning general-purpose, domain-specific, multilingual, and modernized variants.

3.3.1. General-purpose encoder

The study examined multiple general-purpose encoder-only models for text representation. Specifically, BERT, RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2021) were employed. In each case, the standard base configuration was utilized to process caption data.

3.3.2. Domain-specific encoder

HateBERT: HateBERT (Caselli et al., 2021), a model pre-trained on English abusive-language corpora obtained from banned Reddit communities, was included to assess the effectiveness of a domain-specific encoder.

3.3.3. Multilingual encoder

MuRIL: MuRIL (Khanuja et al., 2021), pre-trained on 16 Indian languages and English, was included because of its outstanding results for hate speech detection in Devanagari script languages (Ale et al., 2025).

3.3.4. Modernized encoder

ModernBERT: ModernBERT (Warner et al., 2025), trained on 2 trillion tokens, optimizing the original BERT with modern architectural enhancements was incorporated to assess the impact of large-scale pre-training on caption-based text classification.

3.4. Comparison Baselines

To isolate the contribution of context-aware image captions, the following comparison baselines were established:

3.4.1. Prompting-based approach

Zero-shot prompting was used to establish an initial baseline using the Gemini 3 Pro model (Google DeepMind, 2025b), which demonstrated the strong image understanding capabilities among other Gemini variants (Comanici et al., 2025). The full structured prompt is provided in Appendix B.

3.4.2. Text-only modality

A text-only modality where two multilingual encoder-only transformer models, MuRIL and XLM-RoBERTa-base (Conneau et al., 2019), were evaluated using the OCR-extracted Devanagari meme text.

3.4.3. Vision-text modality

To incorporate visual information, an early fusion technique was implemented in which input images were encoded using Vision Transformer (ViT) (Dosovitskiy et al., 2021), while multilingual encoder-only transformers (Khanuja et al., 2021; Pires et al., 2019; Pudasaini et al., 2023) were used to encode the OCR-extracted text. The obtained image and text representations were concatenated and passed to a linear classification head.

An image-caption modality was also explored to assess whether the encoded image representation provides additional discriminative information beyond the Gemini caption.

3.5. Hyperparameters and Training Configuration

- **Hardware:** NVIDIA GeForce RTX 3090 GPU (24GB VRAM)
- **Optimizer:** AdamW (lr=2e-5, weight decay=0.01, β_1=0.9, β_2=0.999, ε=1e-8)
- **Max sequence length:** 512
- **Batch Size:** 16
- **Epoch:** 8 (early stopping)
- **Random seed:** 0

4. Results and Discussion

The Shared Task was hosted on the **Codabench** [2] platform, where predictions were submitted as binary output files (0 or 1). The submissions were evaluated using macro-averaged precision, recall, F1-score, and accuracy, with leaderboard rankings based on macro F1-score. Table 3 reports the computed metrics using the test split across all modalities, including their respective IDs.

4.1. Zero-shot prompting

As a baseline (see Table 3), the Gemini 3 Pro, chosen for its competitive image understanding capabilities (Comanici et al., 2025; Google DeepMind, 2025b), was used under a zero-shot prompting paradigm (M1), providing no training examples and relying entirely on the model's pre-trained knowledge. This approach yielded the weakest performance across all configurations, with a macro F1-score of 0.3728, precision of 0.3187, recall of 0.4485, and accuracy of only 0.3060. These results highlight the limitations of relying on a large general-purpose VLM without domain-specific fine-tuning and confirm that task-specific training is essential for hate speech detection in Nepali memes.

4.2. Text-only models

Two multilingual text encoders, namely MuRIL (M2) and XLM-RoBERTa-base (M3), were fine-tuned on OCR-extracted meme text. Both models achieved identical test-set performance, with macro F1-score, precision, recall, and accuracy of 0.4018, 0.3358, 0.5000, and 0.6716, respectively. The low macro F1-score indicates that the text-only modality is insufficient, motivating the exploration of multimodal strategies such as text–image fusion.

4.3. Vision-text models

A multimodal strategy combining text and image representations using an early fusion technique was investigated to combine both textual and visual information. All evaluated combinations, including MuRIL + ViT (M4), NepaliBERT + ViT (M5), and mBERT + ViT (M6), significantly outperformed the text-only baseline, achieving F1-scores ranging from 0.6199 to 0.6266, with M4 yielding the best performance.

4.4. Caption-based models

The competitive performance of the vision–text modality motivated the exploration of a caption-based experiment. The experiments were con-

[2] https://www.codabench.org/competitions/12090/

Category	ID	Text Model	Vision Model	Augmentation Type	F1 Score	Precision	Recall	Accuracy
Zero-Shot prompting	M1	Gemini 3 Pro		None	0.3728	0.3187	0.4485	0.3060
Text Only	M2	MuRIL	–	None	0.4018	0.3358	0.5000	0.6716
	M3	XLM-RoBERTa-base	–	None	0.4018	0.3358	0.5000	0.6716
Text + Image	M4	MuRIL	ViT	None	0.6266	0.6535	0.6210	0.7015
	M5	NepaliBERT	ViT	None	0.6194	0.6670	0.6149	0.7090
	M6	mBERT	ViT	None	0.6119	0.6692	0.6091	0.7090
Caption Only	M7	MuRIL	–	None	0.4242	0.3684	0.5000	0.7368
	M8	HateBERT	–	None	0.5991	0.6094	0.6240	0.6119
	M9	BERT-base-uncased	–	None	0.6511	0.6521	0.6503	0.6940
	M10	RoBERTa-base	–	None	0.6694	0.6949	0.6606	0.7313
	M11	DeBERTa-base	–	None	0.4453	0.6730	0.5172	0.6791
	M12	ModernBERT-base	–	None	0.5213	0.5405	0.5520	0.5563
	M13	RoBERTa-base	–	Stochastic	0.7288	0.7241	**0.7472**	0.7463
	M14	**RoBERTa-base**	–	**TTA**	**0.7397**	**0.7612**	0.7285	**0.7836**
Caption + Image	M15	RoBERTa-base	ViT	–	0.6059	0.6553	0.6035	0.7015

Table 3: Performance of all experimental configurations on the CHiPSAL 2026 Subtask A test set. All metrics are macro-averaged. **Bold** = best value per column. TTA = Test-Time Augmentation.

ducted under two settings: with and without stochastic caption augmentation.

4.4.1. Without Stochastic Caption Augmentation

Six models were evaluated: MuRIL (M7), HateBERT (M8), BERT-base-uncased (M9), RoBERTa-base (M10), DeBERTa-base (M11), and ModernBERT-base (M12). Among these, RoBERTa-base performed the best, achieving an F1-score of 0.6694 without augmentation. MuRIL performed the worst with an F1-score of 0.4242, suggesting that multilingual models are less effective for caption-based classification. Despite being trained on domain-specific data or using modernized architectures, HateBERT and ModernBERT-base underperformed, with F1-scores of 0.5991 and 0.5213, respectively, compared to the vision–text modality. This is likely due to domain discrepancies and the limited amount of fine-tuning data available for this low-resource Nepali task.

Additionally, the Caption + Image modality was evaluated using RoBERTa-base + ViT (M15), achieving an F1-score of 0.6059, which is lower than the best-performing RoBERTa-base model under the caption-only category.

4.4.2. With Stochastic Caption Augmentation

Caption-level data augmentation strategies leveraging the stochastic nature of Gemini variants (Comanici et al., 2025; Google DeepMind, 2025a,b) were studied. In this setting, multiple captions were generated per image by sampling at **temperature 1.0**, effectively doubling the minority non-hate class training samples. The RoBERTa-base model trained with this augmentation outperformed the model without augmentation, achieving an F1-score of 0.7288. Applying TTA further improved the F1-score to 0.7397 and secured third place on the CHiPSAL 2026 final leaderboard.

5. Conclusion and Future Work

This study highlights the potential of contextually rich captions in multimodal hate speech detection in the Devanagari script. The systematic evaluation of unimodal (text-only), multimodal (image-text), and caption-based modeling with stochastic caption augmentation and TTA demonstrated the effectiveness of caption-based modeling, achieving **third** place on the CHiPSAL 2026 shared task leaderboard.

Future work may explore contrastive pre-training on large-scale Devanagari meme datasets to align images with context-aware captions using a contrastive learning framework (Chen et al., 2020).

6. Limitations

Despite strong image understanding capabilities, Gemini and its variants (Comanici et al., 2025; Google DeepMind, 2025a,b) lack sufficient knowledge of the Nepali cultural context, idiomatic expressions, and nuanced humor. Over 75% of the generated captions begin with phrases such as "The image is...", "This meme features...", or "The image displays". Such redundant lexical patterns across the dataset may have impacted the learned latent representations. The example memes and generated captions are provided in Appendix C.

7. Acknowledgements

The author thanks the organizers of the CHiPSAL 2026 shared task for providing the annotated datasets and organizing the competition.

8. Bibliographical References

Prabhat Ale, Anish Thapaliya, and Suman Paudel. 2025. MDSBots@NLU of Devanagari script languages 2025: Detection of language, hate speech, and targets using MURTweet. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 308–313, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1994–2003.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (ALW), co-located with EMNLP 2021*, pages 17–25. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Google DeepMind. 2025a. Gemini 3 flash: frontier intelligence built for speed. https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/. Official Google AI blog post. Accessed: 2026-03-01.

Google DeepMind. 2025b. Gemini 3 pro model card. Technical report, Google DeepMind. Model card for Gemini 3 Pro. Accessed: 2026-03-01.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. MuRIL: Multilingual representations for Indian languages. *arXiv preprint arXiv:2103.10730*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624. Curran Associates, Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nikita Moshkov, Botond Mathe, Attila Kertesz-Farkas, Reka Hollandi, and Peter Horvath. 2020. Test-time augmentation for deep learning-based cell segmentation on microscopy images. *Scientific reports*, 10(1):5068.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Shushanta Pudasaini, Subarna Shakya, Aakash Tamang, Sajjan Adhikari, Sunil Thapa, and Sagar Lamichhane. 2023. Nepalibert: Pre-training of masked language model in nepali corpus. In *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 325–330.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Siddhant Bikram Shah, Kritesh Rauniyar, Shuvam Shiwakoti, Surabhi Adhikari, Hariram Veeramani, Kristina T. Johnson, Ali Hurriyetoglu, Hristo Tanev, and Usman Naseem. 2025a. Multimodal hate, humor, and stance event detection in marginalized sociopolitical movements. In *Proceedings of the 8th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Texts*, pages 20–31, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025b. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025c. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547.

A. Image Caption Generation Prompt

(1) **Visual and textual content** — Describe all visible elements and transcribe any embedded Devanagari or English text verbatim.

(2) **Target and framing** — Identify the subject of the meme and note whether any person or group is portrayed with respect to gender, race, religion, caste, or organizational status.

(3) **Tone and intent** — Describe the emotional register, noting whether the content expresses affection, humor, support, criticism, mockery, insult, or glorification of violence, and whether any humor appears benign or malicious in intent.

(4) **Cultural context** — Identify culturally specific symbols, political allusions, or references relevant to the Nepali context.

Do not classify the meme; only describe what is present.

B. Zero-shot Hate Speech Detection Prompt

You are an expert content moderator for Nepali social media content. Analyze the provided meme and classify it according to the exact annotation guidelines used in the NeMeme dataset.

Binary Classification Task

Label 0 (Non-Hate): The meme does NOT contain hate speech. Label 1 (Hate): The meme contains hate speech.

Definition of Hate Speech (Per NeMeme Annotation Guidelines)

A meme is HATEFUL (1) if it meets ANY of these criteria:

- Vilifies, denigrates, bullies, insults, or mocks a subject - Targets based on protected characteristics: gender, race, religion, caste, or organizational status - Contains hateful symbols - Glorifies violence

A meme is NON-HATE (0) if it contains:

- Affection, gratitude, support, motivation - Humor without malicious intent - Constructive criticisms (even if critical of public figures or policies).

C. Example Memes and Generated Captions



Caption: *This image is a satirical side-by-side comparison of two titan figures in Nepalese politics, centered on a linguistic pun regarding their names and the quest for political power. On the left, former Prime Minister K.P. Sharma Oli is shown with a stern expression and a raised hand as if making a point or refusing a request; the Nepali text above him reads (*Dinna Ba*), which translates to "I won't give, Father." On the right, fellow multi-term former Prime Minister Sher Bahadur Deuba is pictured smiling with his hands in a traditional *Namaste* gesture, positioned under the text (*Deu Ba*). The humor stems from the wordplay on the word "Ba" (a respectful term for "Father" used by supporters for both men) and their surnames. "Deu Ba" translating to a plea of "Give, Father," whereas "Dinna Ba" portrays Oli as the obstinate leader refusing to yield.*



Caption: *This humorous Nepali meme plays on a common childhood myth regarding the dangers of swallowing gum. The top text in Nepali translates to "Swallowed a lot of chewing gum when I was young," referencing the widespread (and false) childhood fear that doing so could be fatal or cause permanent damage. The visual features two men in stylish attire, with the focal point being a man on the right wearing a cowboy hat and sunglasses, gesturing in a cool, nonchalant manner. Centered over the scene is the phrase "THUG LIFE" accompanied by sunglasses emojis and various "SARCASM Nepal" watermarks. The bottom text delivers the punchline: "But well, it seems I haven't died even until today."*



Caption: *This informational graphic is divided into two equal vertical halves, used to clarify Nepali terminology for different forms of rice. The left side features a bold red header containing white Nepali text that translates to "Let's call cooked rice 'Bhaat'.". The right side of the graphic has a solid blue header with white Nepali text that translates to "Call puffed rice 'Bhuja'." Together, the two sides serve as a linguistic guide, advocating for the specific use of the word "Bhaat" for steamed rice and "Bhuja" specifically for puffed rice.*



Caption: *This Nepali-language meme humorously references the devastating 2015 earthquake in relation to a popular television character. At the top, bold text translates to "In such a massive earthquake," setting a serious backdrop for an ironic question. On the left side of the image is a classic black-and-white "thinking face" internet meme character, looking upward inquisitively with its hand on its chin. To the right is a photograph of the character "Padey" from the well-known Nepali sitcom *Bhadragol*, recognizable by his signature grey beanie with pointed ears, thick-rimmed glasses, and white-painted beard. The bottom caption asks, "I wonder if Padey Solti's 'Cottage' survived or not?" This is a joke directed at fans of the show, as Padey frequently and boastfully refers to his very humble, ramshackle hut as a "Cottage," leading the meme creator to wonder if such a flimsy structure could have possibly withstood the massive tremors.*

EthosAI@CHiPSAL2026:Hate and Sentiment Understanding in Low-Resource Memes using a Multimodal Approach

Vinayak Bansal[1], **Deepawali Sharma**[2],**Aakash Singh**[1], **Vivek Kumar Singh**[1]
[1]Department of Computer Science, University of Delhi, Delhi, India
[2]School of Computer Science Engineering and Technology (SCSET),
Bennett University, Greater Noida, India
bansal.vinayak2002@gmail.com, deepawali121@gmail.com,
asingh@cs.du.ac.in, vivek@cs.du.ac.in

Abstract

Memes have become a popular way for people to share opinions and emotions on social media, but they are also often used to spread hate and negative sentiments. In this paper, we present our multimodal approach to the CHiPSAL 2026 shared task on multimodal hate and sentiment detection in Nepali memes, which includes two subtasks: hate detection and sentiment analysis. Since memes usually combine both text and images, we first experimented with different unimodal models for text and images separately. After identifying the top two best-performing text and image models, combined them using different fusion techniques. The results show that multimodal models outperform unimodal ones, highlighting that both textual and visual information are important for understanding the context of memes. The multimodal model, which combines sentence-transformers/LaBSE for text and ResNet-18 for image using weighted Fusion technique, achieved a macro F1 score of 0.6614 for Subtask A and sentence-Transformers/LaBSE for text and deit-Base for image using simple Fusion technique, achieved a macro F1 score of 0.4839 for SubTask B, on the test dataset.

Keywords: Hate speech Detection, Low-Resource language, Meme Classification, Multimodal Data, Sentiment analysis

1. Introduction

In the era of social media, memes have become a powerful medium for expressing public sentiment within online communities. For teenagers in particular, memes play a central role in communication.According to a report by YPulse (2019)[1], 75% of individuals between the ages of 13 and 36 share memes. Memes often serve as a means of spreading joy and happiness. However, some people misuse memes to spread hate, especially against vulnerable communities (Mukhtar et al., 2024). Such hateful content can contribute to social unrest and may even lead to violence (Parihar et al., 2021; Bhandari et al., 2023; Singh et al., 2026). Several studies (Kiela et al., 2020; Arya et al., 2024; Das et al., 2020) have focused on detecting hate speech in English- language memes. In contrast, only a limited number of studies (Singh et al., 2024; Nagaraju and Shashirekha, 2025), address hate detection in memes in low-resource languages, and for some languages, no prior work exists. Nepali is one such low-resource language(Sharma et al., 2025). Nepali is the official and most widely spoken language of Nepal, with approximately 19 million native speakers worldwide[2].

The CHiPSAL 2026 task (Thapa et al., 2026) is introduced to specifically focus on detecting hate in this low-resource language setting. The task is divided into two subtasks: (A) identifying hate in memes and (B) identifying sentiment expressed in memes. This task aims to bridge the gap in hate detection and sentiment analysis for the Nepali language. Preventing the spread of hateful content on social media is essential for creating safer and healthier digital platforms for people from all backgrounds.

In this paper, we analyze multimodal architectures and show that both textual and visual information are essential for accurately identifying hate and sentiment in memes. Since memes combine images and text to convey meaning, understanding their full context requires integrating both modalities. To examine this, we evaluate several state-of-the-art (SOTA) models for text and image processing, apply different fusion strategies, and compare their performance. The results demonstrate that multimodal approaches outperform unimodal models for both subtasks.

The subsequent sections of this paper are structured as follows. Section 2 reviews the relevant literature on hateful meme detection and sentiment analysis. Section 3 presents the dataset description. Section 4 describes implemented unimodal and multimodal architectures. Section 5 outlines the experimental setup, while Section 6 discusses the results. Finally, Section 7 concludes the paper

[1]YPulse report 3 Stats That Show What Memes Mean to Gen Z & Millennials https://www.ypulse.com/article/2019/03/05/3-stats-that-show-what-memes-mean-to-gen-z-millennials/

[2]"Nepali language," Wikipedia, last modified 2026, https://en.wikipedia.org/wiki/Nepali_language.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 316–323
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

by summarizing our key contributions and insights.

2. Related Work

With the rapid growth of social media platforms, the need to monitor and detect harmful online content has become increasingly important. Early research in hate speech detection has focused primarily on text-based analysis; studies utilized natural language processing for the identification of hate speech in comments, posts, etc. (Schmidt and Wiegand, 2017; Davidson et al., 2017; Fortuna and Nunes, 2018; Sharma et al., 2024). Studies have been done on the emotional connection of speech, which laid the foundation for sentiment analysis (Plaza-Del-Arco et al., 2021; Nandwani and Verma, 2021; Sadat et al., 2022; Singh et al., 2025). These works demonstrate the interlink between hate speech and sentiment analysis, since negative emotions frequently precede hate speech.

With the rapid rise of internet memes, there has been a shift toward multimodal analysis, as memes usually combine both images and text to convey meaning. Understanding them through only one modality is often not enough. The Hateful Memes Challenge (Kiela et al., 2020) was one of the early efforts that highlighted this issue, showing that memes require models to jointly interpret visual and textual information due to their complex and subtle nature. Later developments such as CLIP further emphasized the importance of aligning images and text representations for better cross-modal understanding (Arya et al., 2024) . Other studies have also explored contextual meme analysis for hate detection, though many of these efforts remain preliminary (Das et al., 2020). While most early research focused on English memes, recent work has started paying attention to code-mixed content, especially Hindi–English memes. These studies suggest that for low-resource languages to benefit from multimodal models, linguistic diversity and language mixing must be considered (Singh et al., 2024). Therefore, there is a need to develop resources that include diverse Indian languages, helping create safer and more inclusive social media environments (Nagaraju and Shashirekha, 2025).

The integration of Large Language Models (LLMs) represents a significant paradigm shift in computational social science, offering advanced reasoning capabilities for complex content analysis (Thapa et al., 2025a). Current frameworks leverage prompt-based approaches and retrieval-augmented methods to handle the scarcity of data and the linguistic intricacies of code-mixed, low-resource memes (Thapa et al., 2025b,c). Despite these advancements, a comprehensive survey of South Asian languages reveals that no research in Nepali specifically concerning meme-based hate and sentiment detection (Sharma et al., 2025). This gap highlights the need for a multimodal dataset for hate detection in the Nepali language. The CHiPSAL2026 (Thapa et al., 2026) shared task addresses this issue by introducing a dataset for detecting hate and analyzing sentiment in low-resource Nepali memes . In this paper, we analyze multimodal models to detect the hate and sentiment analysis within the Nepali memes.

3. Dataset Description

The dataset used in this study is provided by the shared task on Multimodal Hate and Sentiment Understanding in Low-Resource Memes (Thapa et al., 2026) in the CHiPSAL 2026 workshop (Sarveswaran et al., 2026). The task consists of two subtasks. SubTask A is for the identification of Hateful Memes in Nepali. SubTask B is for sentimental analysis of Hateful Memes in Nepali.

3.1. SubTask A

This Subtask is for the identification of Hateful Memes in Nepali. It consists of 1068 memes for training (720 of Hate and 348 of Non Hate), 133 memes for validation (98 of Hate and 35 of Non Hate), and 134 memes for testing, as shown in Table 1.

Table 1: Dataset distribution for SubTask A

Class	Train	Val	Test
Hate	720	98	-
Non-Hate	348	35	-
Total	1068	133	134

3.2. SubTask B

This SubTask is for sentimental analysis of Hateful Memes in Nepali. It consists of 1061 memes for training (341 Negative, 473 Neutral, and 247 Positive), 133 memes for validation (39 Negative, 65 Neutral, and 29 Positive), and 133 memes for testing, as shown in Table 2.

Table 2: Dataset distribution for SubTask B

Sentiment	Train	Val	Test
Negative	341	39	-
Neutral	473	65	-
Positive	247	29	-
Total	1061	133	133

4. Methodology

The dataset consists of memes comprising two modalities: image and text. In this section, we in-

troduce the methodology used in the study to handle both modalities and their fusion. The problem statement is defined as follows:

$$I \in \mathbb{R}^{3\times224\times224}$$
$$T \in \mathbb{R}^{1\times256}$$

where I is a 3-channel image of height 224 and width 224, and T is a 256-dimensional embedding representation.

Subtask A is defined as:

$$y = \begin{cases} 0 & \text{Hate} \\ 1 & \text{Non-Hate} \end{cases}$$

The classifier HC_1 is defined as:

$$HC_1 : (I, T) \rightarrow y, \quad y \in \{0, 1\}$$

Subtask B is defined as:

$$y = \begin{cases} 0 & \text{Negative Sentiment} \\ 1 & \text{Neutral Sentiment} \\ 2 & \text{Positive Sentiment} \end{cases}$$

The classifier HC_2 is defined as:

$$HC_2 : (I, T) \rightarrow y, \quad y \in \{0, 1, 2\}$$

4.1. Unimodal Model

In the unimodal approach, the classifier is designed to process output given only one of the two modalities (e.g., either image or text). This individual contribution serves as a baseline for each modality.

4.1.1. Image

In this section, we summarize the image model analysis. We test models $I_1, I_2, \ldots, I_{10}$ as shown in the image model Table 3. We select the top two individual image models, I_1 and I_2, to be tested further for fusion, as shown in section 4.2.

Table 3: Abbreviations of Image Model Names

Tag	Model Name
I1	Deit-Base
I2	Resnet18
I3	Beit-Base
I4	Resnet50
I5	Vgg16
I6	Efficientnet-B4
I7	Efficientnet-B0
I8	Convnext-Base
I9	Vit-Large
I10	Vit-Base

1. I_1**: DeiT-Base** is a transformer-based model introduced by (Touvron et al., 2021). This model follows the Vision Transformer pipeline with a key modification: knowledge distillation through a distillation token. In this model, an additional distillation token is introduced, which learns from a CNN teacher model. Here, the model is supervised by both the ground truth and the teacher predictions. This allows the model to perform better on smaller datasets where data is limited for parameter tuning.

2. I_2**: ResNet-18** is a convolution-based (CNN) model introduced by (He et al., 2016). This model handles the degradation problems in CNNs with the help of the residual connection method, which is mathematically expressed as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\}) + \mathbf{x}$$

where $\mathbf{x}$ is the input and $\mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\})$ represents the output of the residual mapping. This residual network helps reduce the vanishing gradient problem, enabling the model to efficiently handle degradation without additional parameters, which helps the model train and perform better on smaller datasets.

4.1.2. Text

In this section, we summarize the text extraction from images and the text models. For text extraction, we use EasyOCR and the Qwen model. EasyOCR text was provided by CHiPSAL; we also use Qwen2-VL-7B-Instruct to generate image descriptions alongside the image OCR. As neither EasyOCR nor Qwen OCR is fully reliable on its own, we extracted text from both to improve overall OCR accuracy. Beyond OCR, the Qwen model is used to generate further details of the image. Listings 1 and 2 provide the prompt templates for Subtask A and Subtask B, respectively.

```
messages = [
    {
        "role": "user",
        "content": [
            {"type": "image", "image": image_path},
            {"type": "text", "text": (
                "Task: 1. OCR Nepali/English text. "
                "2. Describe visual/textual meaning. "
                "3. Classify: 'Hate' or 'Not Hate'. "
                "4. Provide justification.\n"
                "Output: OCR_TEXT, MEANING, LABEL, JUSTIFICATION"
            )}
        ]
    }
]
```

Listing 1: Prompt Template for Subtask A: Hate Speech Detection

```
messages = [
    {
        "role": "user",
        "content": [
            {"type": "image", "image": image_path},
            {"type": "text", "text": (
                "Task: 1. OCR Nepali/English text. "
                "2. Describe meme meaning. "
                "3. Determine Sentiment: Positive, Neutral, or Negative. "
                "4. Explain sentiment reason.\n"
                "Output: OCR_TEXT, MEANING, SENTIMENT, SENTIMENT_REASON"
            )}
        ]
    }
]
```

Listing 2: Prompt Template for Subtask B: Sentiment Analysis

Table 4: Abbreviations of Text Model Names

Tag	Model Name
T1	Sentence-Transformers/LaBSE
T2	Bert-Base-Multilingual-Cased
T3	Distilbert-Base-Multilingual-Cased
T4	Sentence-Transformers/Paraphrase-Multilingual-Mpnet-Base-V2
T5	Sentence-Transformers/Paraphrase-Multilingual-MiniLM-L12-V2
T6	Xlm-Roberta-Base
T7	Ai4Bharat/Indic-Bert
T8	Microsoft/Mdeberta-V3-Base
T9	Google/Muril-Base-Cased
T10	Google/Canine-C

We test models $T_1, T_2, \ldots, T_{10}$ as shown in the text model Table 4. We select the top two individual text models, T_1 and T_2, to be tested further for fusion in section 4.2.

- T_1**: Sentence-Transformers/LaBSE** is a multilingual sentence embedding model introduced by (Feng et al., 2022). This model is based on Multilingual BERT (mBERT) with 12 transformer layers. The model uses a shared semantic vector space for different languages; it provides a similar vector space for both the Nepali text and its English translation. This helps in maintaining the complete text constructed in the preceding sections within a similar vector space, allowing the model to better understand the context of the meme.
- T_2**: Bert-Base-Multilingual-Cased (mBERT)** is a multilingual language model trained on Wikipedia data from 104 languages. This model aligns Nepali and English using cross-lingual alignment, which happens implicitly through the shared training process. The model has 12 transformer layers, and its cross-lingual alignment helps it work on unlabeled Nepali data based on fine-tuning on English.

4.2. Multimodal Model

In the previous subsection 4.1, we established a baseline for how image and text performed individually. Based on a single modality, meme content cannot always be reliably classified as hate or non-hate, nor can we perform sentiment analysis; therefore, we perform different types of fusion on the top two performing image models (I_1, I_2) and the top two performing text models (T_1, T_2).

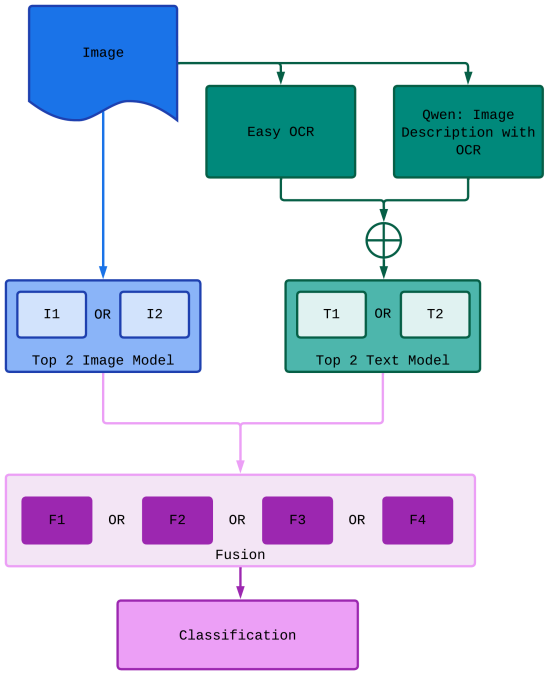


Figure 1: Proposed Multimodal Design

Figure 1 presents our proposed multimodal design. In the fusion of the two modalities, the modalities are first projected into a common dimension. Let $\mathbf{t} \in \mathbb{R}^{d_t}$ denote the text feature vector and $\mathbf{v} \in \mathbb{R}^{d_v}$ denote the image feature vector.

$$\tilde{\mathbf{t}} = W_t \mathbf{t} + b_t, \quad W_t \in \mathbb{R}^{256 \times d_t}, \quad b_t \in \mathbb{R}^{256} \tag{1}$$

$$\tilde{\mathbf{v}} = W_v \mathbf{v} + b_v, \quad W_v \in \mathbb{R}^{256 \times d_v}, \quad b_v \in \mathbb{R}^{256} \tag{2}$$

Such that

$$\tilde{\mathbf{t}}, \tilde{\mathbf{v}} \in \mathbb{R}^{256}.$$

- F_1: **Simple Fusion** In this fusion, the projected vectors $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{v}}$ are concatenated and passed through a fully connected layer for classification.

$$\mathbf{x} = [\tilde{\mathbf{v}}; \tilde{\mathbf{t}}] \in \mathbb{R}^{512} \tag{3}$$

  The classifier is defined as:

$$\hat{y} = \sigma(W_c \mathbf{x} + b_c) \tag{4}$$

  where $W_c \in \mathbb{R}^{1\times512}, b_c \in \mathbb{R}$.

- F_2: **Weighted Fusion** In weighted fusion, we introduce learnable scalar weights α and β:

$$\mathbf{x} = \alpha\tilde{\mathbf{t}} + \beta\tilde{\mathbf{v}}, \quad \mathbf{x} \in \mathbb{R}^{256} \tag{5}$$

  where $\alpha, \beta \in \mathbb{R}$ are trainable parameters. The classifier is defined as:

$$\hat{y} = \sigma(W_c \mathbf{x} + b_c), \quad W_c \in \mathbb{R}^{1\times256}. \tag{6}$$

- F_3: **Cross-Attention Fusion** We employ a single-head cross-attention mechanism with text features as the query and image features as the key and value:

$$Q = \tilde{\mathbf{t}}, \quad K = \tilde{\mathbf{v}}, \quad V = \tilde{\mathbf{v}} \tag{7}$$

  The attention mechanism is defined as:

$$\mathbf{x} = \text{softmax}\left(\frac{QK^\top}{\sqrt{256}}\right)V \tag{8}$$

  where $\mathbf{x} \in \mathbb{R}^{256}$. The final prediction is:

$$\hat{y} = \sigma(W_c \mathbf{x} + b_c), \quad W_c \in \mathbb{R}^{1\times256}. \tag{9}$$

- F_4: **Element-Wise Fusion** Element-wise fusion is performed via direct addition:

$$\mathbf{x} = \tilde{\mathbf{t}} + \tilde{\mathbf{v}}, \quad \mathbf{x} \in \mathbb{R}^{256} \tag{10}$$

  The classifier output is:

$$\hat{y} = \sigma(W_c \mathbf{x} + b_c), \quad W_c \in \mathbb{R}^{1\times256}. \tag{11}$$

5. Experimental Setup

In this section, we provide the details of the experimental setup used in Section 4 for model training. All experiments are performed using an NVIDIA RTX 5000 Ada (32 GB GPU VRAM) paired with 256 GB system RAM and an Intel Xeon W7-2475X processor as the hardware platform. The primary software libraries employed for this study are PyTorch, Pandas, and OpenCV.

Table 5 summarizes the hyperparameters used in our experiments. All models are trained using a batch size of 8 to balance performance and memory usage. Training is performed for a maximum of 50 epochs. To avoid overfitting, early stopping is applied based on the validation loss, with a patience value of 5. We use ReduceLROnPlateau as the learning rate scheduler, with a reduction factor of 0.5 and a patience of 2. For Subtask A, BCEWithLogitsLoss is used as the loss function. For Subtask B, CrossEntropyLoss is applied. The AdamW optimizer is used to update the model parameters during training.

Table 5: Training Hyperparameters

Hyperparameter	Value
Batch Size	8
Epochs	50
Early Stopping Patience	5
Learning Rate	1×10^{-5}
Weight Decay	1×10^{-4}
Optimizer	AdamW
Scheduler	ReduceLROnPlateau
Scheduler Factor	0.5
Scheduler Patience	2
Loss Function	BCEWithLogitsLoss / CrossEntropyLoss

6. Results

Table 6 presents the performance of unimodal models on the validation dataset. T_1 and T_2 are the best-performing text models. T_1 achieves the best validation macro F1 score of 0.605 for Subtask A and 0.4443 for Subtask B, followed by T_2 with a validation macro F1 score of 0.597 for Subtask A and 0.4342 for Subtask B. For image models, I_1 and I_2 are the top performers. I_1 achieves the best validation macro F1 score of 0.5735 for Subtask A and 0.4606 for Subtask B, followed by I_2 with a validation macro F1 score of 0.5929 for Subtask A and 0.4525 for Subtask B.

Table 7 presents the results for the multimodal models on the validation dataset. The results indicate that for Subtask A, the top-performing multimodal model is F_3 using T_1 and I_1 (macro F1 of 0.6568), followed by F_3 using T_1 and I_2 (macro F1 of 0.6502), and F_2 using T_1 and I_2 (macro F1 of 0.6425). For Subtask B, the top-performing multimodal model is F_2 using T_1 and I_1 (macro F1 of 0.4755), followed by F_3 using T_1 and I_1 (macro F1 of 0.4747), and F_1 using T_1 and I_1 (macro F1 of 0.4613).

We select the top three performing multimodal models on the validation dataset for evaluation on the final test dataset. Table 8 provides the results for the top three models on the test dataset; notably, F_2 with the T_1 and I_2 model performs the best with a macro F1 of **0.6614**. Table 8 provides the results

for the test dataset, where the F_1 fusion on the T_1 and I_1 model performs the best with a macro F1 of **0.4839**.

We attempted to detect hate speech and analyze meme sentiment based on the direct output of the Qwen model; however, the model was unable to successfully classify all the memes. Consequently, we were required to pass the Qwen descriptions along with the OCR to the text models. A closer look at the results reveals a bottleneck caused by the count of randomly initialized learnable parameters introduced in the fusion layers. The results make it evident that the model is unable to completely train these randomly initialized fusion parameters.

Simple and weighted fusion methods are performing better than cross-attention-based fusion. For Subtask A, weighted fusion performs the best; however, for Subtask B, which has a smaller dataset size per class, simple fusion is performing better than even weighted fusion. These results demonstrate that for a smaller dataset, the introduction of untrained parameters results in the addition of more noise. We also observe that element-wise fusion leads to a loss of essential information, resulting in lower performance than unimodal approaches. The model is unable to retain the unimodal information in that configuration. Hence, simple concatenation, providing a balance between dataset size and the introduction of new parameters, yields the best performing results.

Table 6: Performance Comparison of Unimodal Models (Validation Dataset)

Mod.	Tag	Model Name	Sub A (F1)	Sub B (F1)
Text (Unimodal)				
Text	**T1**	**Sentence-Transformers/ LaBSE**	**0.605**	**0.4443**
Text	**T2**	**Bert-Base-Multilingual-Cased**	**0.597**	**0.4342**
Text	T3	DistilBert-Base-Multilingual-Cased	0.5535	0.4241
Text	T4	Sent.-Trans./ Para.-Multiling.-Mpnet-Base-V2	0.562	0.4057
Text	T5	Sent.-Trans./ Para.-Multiling.-MiniLM-L12-V2	0.5793	0.3921
Text	T6	XLM-Roberta-Base	0.4242	0.2189
Text	T7	Ai4Bharat/ Indic-Bert	0.4242	0.324
Text	T8	Microsoft/ MDeBerta-V3-Base	0.4242	0.2189
Text	T9	Google/ MuRIL-Base-Cased	0.4242	0.2189
Text	T10	Google/ Canine-C	0.4242	0.2189
Image (Unimodal)				
Image	**I1**	**DeiT-Base**	**0.5735**	**0.4606**
Image	**I2**	**ResNet18**	**0.5929**	**0.4525**
Image	I3	BeiT-Base	0.581	0.4388
Image	I4	ResNet50	0.5822	0.413
Image	I5	VGG16	0.5794	0.3504
Image	I6	EfficientNet_B4	0.5694	0.4485
Image	I7	EfficientNet_B0	0.5427	0.3997
Image	I8	ConvNext_Base	0.5393	0.4371
Image	I9	ViT-Large	0.5654	0.3946
Image	I10	ViT-Base	0.5347	0.4149

Table 7: Performance Comparison of Multimodal Models on Subtasks A and B (Validation Dataset)

Text	Image	Fusion	SubTask A Macro F1	SubTask B Macro F1
T_1	I_1	F_4	0.5186	0.4579
T_1	I_1	F_3	**0.6568**	**0.4747**
T_1	I_1	F_1	0.6171	**0.4613**
T_1	I_1	F_2	0.6252	**0.4755**
T_1	I_2	F_4	0.5395	0.4336
T_1	I_2	F_3	**0.6502**	0.4546
T_1	I_2	F_1	0.6094	0.4365
T_1	I_2	F_2	**0.6425**	0.3946
T_2	I_1	F_4	0.5423	0.3924
T_2	I_1	F_3	0.5574	0.4201
T_2	I_1	F_1	0.5130	0.4275
T_2	I_1	F_2	0.5405	0.3793
T_2	I_2	F_4	0.6094	0.4337
T_2	I_2	F_3	0.5585	0.4301
T_2	I_2	F_1	0.5485	0.3847
T_2	I_2	F_2	0.5970	0.4089

Note: F_1: Simple Fusion, F_2: Weighted fusion, F_3: Element-Wise fusion, F_4: Cross-Attention Fusion.

Table 8: Testing Dataset Results for Multimodal Fusion

Text	Image	Fusion	Macro F1	A	P	R
Subtask A						
T1	I1	F3	0.6229	0.6791	0.6324	0.6275
T1	I2	F3	0.6080	0.6866	0.6321	0.6040
T1	**I2**	**F2**	**0.6614**	**0.7164**	**0.6743**	**0.6553**
Subtask B						
T1	I1	F3	0.4336	0.4586	0.4611	0.4349
T1	**I1**	**F1**	**0.4839**	**0.4962**	**0.4824**	**0.4857**
T1	I1	F2	0.4670	0.5038	0.4975	0.4583

A: Accuracy; **P**: Precision (Macro); **R**: Recall (Macro).

7. Conclusion

Detecting hate and sentiment in memes is important for building safe and inclusive online spaces. Memes spread ideas quickly and often hide harmful intent, which makes automatic detection necessary, especially for low-resource languages like Nepali, where data and tools are limited. In this paper, we presented our multimodal approach to the CHiPSAL 2026 shared task on multimodal hate and sentiment detection in Nepali memes. Various text-based and image-based unimodal models are implemented and evaluated. The top-performing models from both modalities are then combined using different fusion techniques. The multimodal approach outperforms the unimodal approach. The best-performing model combines T_1: Sentence-Transformers/LaBSE for text and I_2: ResNet-18 for image using F_2: Weighted Fusion technique, and the reported macro F1-scores of 0.6614 for Subtask A and T_1: Sentence-Transformers/LaBSE for text and I_1: Deit-Base for image using F_1: Simple Fusion technique, and the reported macro F1-scores of $0.0.4839$ for SubTask B, respectively. This work serves as a valuable baseline and resource for the research community and supports the development of more robust and effective computational tools for multimodal content understanding in low-resource languages.

Code Availability

The code is available at the following link https://github.com/vinayakbansal-2002/EthosAI-Chipsal

Acknowledgements

We acknowledge the financial assistance and academic support provided to one of the authors under the Visvesvaraya PhD Scheme by the Ministry of Electronics and Information Technology, Government of India, in the form of the Visvesvaraya Fellowship.

8. Bibliographical References

Greeshma Arya, Mohammad Kamrul Hasan, Ashish Bagwari, Nurhizam Safie, Shayla Islam, Fatima Rayan Awad Ahmed, Aaishani De, Muhammad Attique Khan, and Taher M Ghazal. 2024. Multimodal hate speech detection in memes using contrastive language-image pre-training. *IEEe Access*, 12:22359–22375.

Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1994–2003.

Abhishek Das, Japsimar Singh Wahi, and Siyao Li. 2020. Detecting hate speech in multi-modal memes. *arXiv preprint arXiv:2012.14891*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 878–891.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *Acm Computing Surveys (Csur)*, 51(4):1–30.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624.

Shahira Mukhtar, Qurat Ul Ain Ayyaz, Sadaf Khan, Atiya Muhammad Nawaz Bhopali, Muhammad Khalid Mehmood Sajid, Allah Wasaya Babbar, et al. 2024. Memes in the digital age: A sociolinguistic examination of cultural expressions and communicative practices across border. *Educational Administration: Theory and Practice*, 30(6):1443–1455.

Rachana Nagaraju and Hosahalli Lakshmaiah Shashirekha. 2025. Towards safer social media: Multimodal hate speech detection in memes across diverse indian languages.

Pansy Nandwani and Rupali Verma. 2021. A review on sentiment analysis and emotion detection from text. *Social network analysis and mining*, 11(1):81.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Flor Miriam Plaza-Del-Arco, M Dolores Molina-González, L Alfonso Ureña-López, and María Teresa Martín-Valdivia. 2021. A multi-task learning approach to hate speech detection leveraging sentiment analysis. *IEEE Access*, 9:112478–112489.

Anwar Sadat, Herman Lawelai, and Ansar Suherman. 2022. Sentiment analysis on social media: Hate speech to the government on twitter. *PRAJA: Jurnal Ilmiah Pemerintahan*, 10(1):69–76.

Kengatharaiyer Sarveswaran, Surendrabikram Thapa, Ashwini Vaidya, Tafseer Ahmed Khan, and Bal Krishna Bal. 2026. Findings of the second workshop on challenges in processing south asian languages (chipsal 2026). In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media*, pages 1–10.

Deepawali Sharma, Tanusree Nath, Vedika Gupta, and Vivek Kumar Singh. 2025. Hate speech detection research in south asian languages: A survey of tasks, datasets and methods. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(3):1–44.

Deepawali Sharma, Aakash Singh, and Vivek Kumar Singh. 2024. Thar-targeted hate speech against religion: A high-quality hindi-english code-mixed dataset with the application of deep learning models for automatic detection. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Aakash Singh, Vinayak Bansal, Muskan Saini, Deepawali Sharma, and Vivek Kumar Singh. 2026. Safeplay-x: A comprehensive gameplay video dataset for violence detection with explainable deep learning applications. *Expert Systems with Applications*, page 131724.

Aakash Singh, Deepawali Sharma, and Vivek Kumar Singh. 2024. Mimic: Misogyny identification in multimodal internet content in hindi-english code-mixed language. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Aakash Singh, Deepawali Sharma, and Vivek Kumar Singh. 2025. Emogif: A multimodal approach to detect emotional support in animated gifs. *IEEE Transactions on Computational Social Systems*.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Surabhi Adhikari, Hariram Veeramani, Mehwish Nasim, and Usman Naseem. 2025a. Large language models (llm) in computational social science: prospects, current state, and challenges. *Social Network Analysis and Mining*, 15(1):4.

Surendrabikram Thapa, Shuvam Shiwakoti, Siddhant Bikram Shah, Kritesh Rauniyar, Laxmi Thapa, Surabhi Adhikari, Kristina T. Johnson, Kengatharaiyer Sarveswaran, Bal Krishna Bal, and Usman Naseem. 2026. Multimodal hate and sentiment understanding in low-resource text-embedded images for online safety and digital well-being. In *Proceedings of the Second Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Hariram Veeramani, Liang Hu, Qi Zhang, Wei Wang, and Usman Naseem. 2025b. A multimodal prompt-based framework for analyzing code-mixed and low-resource memes. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1913–1923.

Surendrabikram Thapa, Hariram Veeramani, Imran Razzak, Roy Ka-Wei Lee, and Usman Naseem. 2025c. Cross platform multimodal retrieval augmented distillation for code-switched content understanding. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2042–2051.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. 2021. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR.

Author Index

ABUL HASNAT, MD., 237
Acharya, Ashish, 244, 258
Adhikari, Abiral, 300
Adhikari, Surabhi, 219
Ahmad, Saad, 49
Ahmed, Kawsar, 92
Ahmed, Tafseer, 1
Anzum, Sabila, 250
Aryal, Pragya, 244
Asif, Muhammad Hassan, 40
Asim, Kulsoom, 59

Bairi, Vennela, 18
Bal, Bal Krishna, 1, 105, 219
Bandara, 198
Bansal, Vinayak, 316

Chowdhury, Alvee Hasan, 237

Debnath, Rotna Dipika, 292
Dhami, Kashish Kumari, 115

Faisal, Adnan, 237
Faisal, Fatima, 9
Fondekar, Ashweta, 229

Ghimire, Rupak Raj, 105
Ghimire, Sadikshya, 154
Gupta, Vibhu, 137
Gurusinghe, Ranidu Hansaka, 68

Hammad, Muhammad, 49
Haque, Farah Binta, 85
Hoque, Mohammed Moshiul, 92
Hossain, Md. Sajjad, 92

Jayatilleke, Nevidu, 68
Johnson, Kristina T, 219

Kabir, Md. Abtahee, 284
Karim, Asim, 49
Karunakaran, Sumirtha, 176
K.C., Bikram, 258
Khadka, Rohit, 244
Khadka, Supriya, 115, 154
KHAN, SUNY MD ASHRAF, 92
Khanal, Nitiz, 275
Khanal, Reewaj, 300
Khanal, Yudhin, 169
Khatiwada, Anish, 244
Krishnamurthy, Parameswari, 18
Kumar, Rakesh, 127
Kumar, Sandesh, 59
Kunwar, Kushal, 169

Labiba, Ayesha, 292
Latif, Kinza, 40

Maharjan, Sujal, 209
Mallik, Arpita, 250, 292
Mansoor, Adina Adnan, 9
Moni, Joyeta Barua, 284
Muntaha, Sidratul, 250
Murad, Hasan, 250, 284, 292

Naseem, Usman, 219
Nasim, Bushra, 40
Nasim, Zarmeen, 40
Nasir, Azkaa, 9
Nerujan, Sathasivam, 190
Nivethiga, Rasan, 176

Orny, Noore Tamanna, 284

Pandey, Ishan, 169
Pandit, Aadarsh, 169
Patel, Suraj, 115
Pawar, Jyoti, 229
Poudel, Jenny, 105
Poudel, Sweta, 209
Pushpananda, Randil, 198

Rafi, Md Shoaib Akhter, 85
Rana, Satyam, 154
Rauniyar, Kritesh, 209, 219
Ravikiran, Manikandan, 137
Regmi, Raunak, 154
Regmi, Sunil, 169, 267
Ruhi, Shahrin Afroz Hoque, 292

Sadeque, Farig, 85
Saha, Shishir, 85

Saleem, Tauqeer, 9
Saluja, Rohit, 137
Samad, Abdul, 9, 59
Sarveswaran, Kengatharaiyer, 1, 176, 190, 219
Shah, Akshyat, 258
Shah, Siddhant Bikram, 219
Sharma, Deepawali, 316
Sharma, Rishikesh Kumar, 105
Sherpa, Norden, 115
Shivolkar, Milind, 229
Shiwakoti, Shuvam, 209, 219
Shrestha, Arju, 105
Shrestha, Astha, 209
Shrestha, Safal Narshing, 105
Shrestha, Sandesh, 258
Shrestha, Sanket, 154
Shrestha, Suman, 267
Singh, Aakash, 316
Singh, Avyaya, 32
Singh, Saugat, 267
Singh, Vivek Kumar, 316
Sivakumaran, Gokulan, 198
Subedi, Bipesh, 267

Thapa, Laxmi, 209, 219
Thapa, Rabin, 209, 258
Thapa, Surendrabikram, 1, 209, 219
Thapaliya, Anish, 308
Tiwari, Rupak, 105
Tiwari, Tanmay, 137
Tu Zahra, Fatima, 59

Ullah, Faizad, 49

Vaidya, Ashwini, 1
Varsha, Jeyarajalingam, 176
Velayuthan, Menan, 176
Vishwakarma, Shashank, 127

Wagle, Samir, 300
Washington, Jonathan N., 32

Yasin, Md, 85
Yousuf, Mahrukh, 9

Zeeshan, Muhammad, 49
Zohair, Muhammad, 40