LREC-COLING 2024

The 7th Workshop on Indian Language Data Resource and Evaluation @LREC-COLING-2024 (WILDRE-7)

Workshop Proceedings

Editors
Girish Jha, Sobha Lalitha Devi, Kalika Bali and Atul Kr. Ojha

25 May, 2024
Torino, Italia

Proceedings of the 7th Workshop on Indian Language Data Resource and Evaluation @LREC-COLING-2024 (WILDRE-7)

ISBN 978-2-493814-37-1
ISSN 2951-2093 (COLING); 2522-2686 (LREC)

Jointly organized by the ELRA Language Resources Association
and the International Committee on Computational Linguistics

Preface

WILDRE – the 7th Workshop on Indian Language Data: Resources and Evaluation is being organized in Torino, Italia on May 25th, 2024 under the LREC-COLING 2024 platform. India has a huge linguistic diversity and has seen concerted efforts from the Indian government and industry towards developing language resources. ELRA Language Resources Association and its associate organizations have been very active and successful in addressing the challenges and opportunities related to language resource creation and evaluation. It is, therefore, a great opportunity for resource creators of Indian languages to showcase their work on this platform and also to interact and learn from those involved in similar initiatives all over the world.

The broader objectives of the 7th WILDRE will be

- to map the status of Indian Language Resources
- to investigate challenges related to creating and sharing various levels of language resources
- to promote a dialogue between language resource developers and users
- to provide an opportunity for researchers from India to collaborate with researchers from other parts of the world

The call for papers received a good response from the Indian language technology community. In addition, WILDRE-7 included a Shared Task on Code-mixed Less-resourced Sentiment Analysis for Indo-Aryan Languages.

This year, we selected only three papers for oral, one findings paper and eight for poster presentations (including two system descriptions and one non-archival).

Workshop Organisers

Workshop Chairs

Girish Nath Jha, Chairman, Commission for Scientific and Technical Terminology, MoE, GOI and JNU, New Delhi
Kalika Bali, Microsoft Research India Lab, Bangalore
Sobha L, AU-KBC, Anna University
Atul Kr. Ojha, University of Ireland Galway, Ireland

Program Committee

Anil Kumar Singh, IIT BHU, Benaras
Anoop Kunchukuttan, Microsoft AI and Research, India
Anupam Basu, Director, NIIT, Durgapur
Arulmozi Selvaraj, University of Hyderabad
Asif Iqbal, IIT Patna, Patna
Atul Kr. Ojha, University of Ireland Galway, Ireland & Panlingua Language Processing LLP, India
Bogdan Babych, Heidelberg University, Germany
Daan van Esch, Google, USA
Dafydd Gibbon, Universität Bielefeld, Germany
Dipti Mishra Sharma, IIIT-Hyderabad
Elizabeth Sherley, IITM-Kerala, Trivandrum
Gaurav Negi, University of Galway
Georg Rehm, DFKI, Germany
Girish Nath Jha, Chairman, Commission for Scientific and Technical Terminology, MoE, GOI and JNU, New Delhi
Jolanta Bachan, Adam Mickiewicz University, Poland
Joseph Mariani, LIMSI-CNRS, France
Khalid Choukri, ELRA, France
Lars Hellan, NTNU, Norway
Manji Bhadra, Bankura University, West Bengal
Malhar Kulkarni, IIT Bombay
Massimo Moneglia, University of Florence, Italy
Monojit Choudhary, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi
Narayan Choudhary, CIIL, Mysore
Niladri Shekhar Dash, ISI Kolkata
Panchanan Mohanty, GLA, Mathura
Priya Rani, University of Galway
Rajeev R R, ICFOSS, Trivandrum
Shantipriya Parida, Silo AI, Finland
Shagun Sinha, Amity University, Noida, India
Shivaji Bandhopadhyay, Jadavpur University
Sobha Lalitha Devi, AU-KBC Research Centre, Anna University
Subhash Chandra, Delhi University
Swaran Lata, Retired Head, TDIL, MCIT, Govt of India
Virach Sornlertlamvanich, Thammasat Univeristy, Bangkok, Thailand
Zygmunt Vetulani, Adam Mickiewicz University, Poland

Table of Contents

Conference Program

Saturday, May 25, 2024

14:00–14:05 ***Welcome by Workshop Chairs***

14:05–15:00 *Keynote Lecture*
TBD

15:00–16:00 **Oral Session-I**

15:00–15:25 *Towards Disfluency Annotated Corpora for Indian Languages*
Chayan Kochar, Vandan Vasantlal Mujadia, Pruthwik Mishra and Dipti Misra Sharma

15:25–15:45 *EmoMix-3L: A Code-Mixed Dataset for Bangla-English-Hindi for Emotion Detection*
Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos and Marcos Zampieri

15:45–16:00 *Findings of the WILDRE Shared Task on Code-mixed Less-resourced Sentiment Analysis for Indo-Aryan Languages*
Priya Rani, Gaurav Negi, Saroj Jha, Shardul Suryawanshi, Atul Kr. Ojha, Paul Buitelaar and John P. McCrae

16:00–16:30 **Coffee break/Poster Session**

16:00–16:30 *Multilingual Bias Detection and Mitigation for Indian Languages*
Ankita Maity, Anubhav Sharma, Rudra Dhar, Tushar Abhishek, Manish Gupta and Vasudeva Varma

16:00–16:30 *Dharmaśāstra Informatics: Concept Mining System for Socio-Cultural Facet in Ancient India*
Arooshi Nigam and Subhash Chandra

16:00–16:30 *Exploring News Summarization and Enrichment in a Highly Resource-Scarce Indian Language: A Case Study of Mizo*
Abhinaba Bala, Ashok Urlana, Rahul Mishra and Parameswari Krishnamurthy

16:00–16:30 *Finding the Causality of an Event in News Articles*
Sobha Lalitha Devi and Pattabhi RK Rao

16:00–16:30 *Creating Corpus of Low Resource Indian Languages for Natural Language Processing: Challenges and Opportunities*
Pratibha Dongare

Saturday, May 25, 2024 (continued)

16:00–16:30 *FZZG at WILDRE-7: Fine-tuning Pre-trained Models for Code-mixed, Less-resourced Sentiment Analysis*
Gaurish Thakkar, Marko Tadić and Nives Mikelic Preradovic

16:00–16:30 *MLInitiative@WILDRE7: Hybrid Approaches with Large Language Models for Enhanced Sentiment Analysis in Code-Switched and Code-Mixed Texts*
Hariram Veeramani, Surendrabikram Thapa and Usman Naseem

16:30–16:55 Oral Session-II

16:30–16:55 *Aalamaram: A Large-Scale Linguistically Annotated Treebank for the Tamil Language*
A M Abirami, Wei Qi Leong, Hamsawardhini Rengarajan, D Anitha, R Suganya, Himanshu Singh, Kengatharaiyer Sarveswaran, William Chandra Tjhi and Rajiv Ratn Shah

16:55–17:40 Panel discussion

17:40–17:50 ***Valedictory Sessioner a title here***

17:50–17:55 ***Vote of Thanks***

Towards Disfluency Annotated Corpora for Indian Languages

Chayan Kochar, Vandan Mujadia, Pruthwik Mishra, Dipti Misra Sharma
LTRC - International Institute of Information Technology Hyderabad
{chayan.kochar, vandan.mu, pruthwik.mishra}@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

In the natural course of spoken language, individuals often engage in thinking and self-correction during speech production. These instances of interruption or correction are commonly referred to as disfluencies. When preparing data for subsequent downstream NLP tasks, these linguistic elements can be systematically removed, or handled as required, to enhance data quality. In this study, we present a comprehensive research on disfluencies in Indian languages. Our approach involves not only annotating real-world conversation transcripts but also conducting a detailed analysis of linguistic nuances inherent to Indian languages that are necessary to consider during annotation. Additionally, we introduce a robust algorithm for the synthetic generation of disfluent data. This algorithm aims to facilitate more effective model training for the identification of disfluencies in real-world conversations, thereby contributing to the advancement of disfluency research in Indian languages.

Keywords: disfluency, annotation guidelines, synthetic augmentation, Indian languages

1. Introduction

Natural speech has its own uniqueness. Written text tends to be very fluent and can be readily used for NLP tasks after preprocessing. In contrast, people often think and speak simultaneously during a discussion when speaking naturally. Individuals often exhibit a reflexive tendency to rectify errors upon recognizing inaccuracies in their speech. This can involve editing, reformulating, or even starting over from scratch. This is a normal, intuitive process that seamlessly gets mixed in spontaneous conversational interactions. Thus natural speech often exhibits such interruptions and disruptions known as disfluencies (Shriberg, 1994).

Disfluencies can be classified into mainly 5 categories: filler words, pet phrases, repetitions, repair and false starts. Though there are other naming conventions or groups which may overlap with the ones mentioned, but there is a distinct characteristic to every disfluent utterance. Every disfluent utterance or a phrase comprises of a ***reparandum***, usually followed by a verbal cue, the *interruption point*, an optional ***edit term***, and finally the optional ***alteration*** (Shriberg, 1994; Heeman and Allen, 1999). The alteration is what an ideal fluent text would have been, replacing the reparandum and editing terms.

These phenomena encompassing hesitations, repeats, corrections etc which are so abundant yet unnoticeable, is what makes the problem so interesting. These disfluent terms can be eliminated because they do not add to the semantics of the sentence, thus producing a noise free data ready to feed to the machines.

This very aspect makes disfluency correction a very crucial factor for any other NLP downstream tasks like MT (Rao et al., 2007; Wang et al., 2010), question answering (Gupta et al., 2021) etc. If the fundamental tasks like these are jeopardized, then all other tasks following them would yield poor output as well. To get this done, a robust set of annotation guidelines is paramount for ensuring the quality, consistency, and reliability of annotated data in any research endeavor, particularly in the field of Natural Language Processing (NLP). A set of detailed annotation guidelines would bring in consistency, reduced ambiguity, scalability and most importantly cross dataset compatibility due to abundance of linguistic features which are common in Indian Languages.

India has a rich linguistic diversity, with about 1369 rationalized mother tongues and numerous more under resourced languages [1]. Given the vast array of linguistic nuances and variations present in India, any NLP-related problem-solving approach must account for this diversity to ensure its effectiveness and applicability within the Indian context. Therefore, it is imperative to prioritize the development of NLP technologies tailored to the specific linguistic landscape of India, facilitating broader accessibility and utility for its diverse population. In light of the lack of good amount of labelled data for Indian languages, the concept of synthetic augmentation becomes much more relevant. Due to the newly created dataset's wide range of variations and scenarios, it not only tackles the issues of data scarcity and class imbalance but also improves model generalisation. Research suggests that this indeed helps in overall performance of the model. (Passali et al., 2022; Kundu et al., 2022)

Extensive research has been conducted on disfluencies (Colman and Healey, 2011; Shriberg,

[1] Census 2011

WILDRE-7 Workshop, pages 1–10

1994) along with work on identification and/or removal of such disfluencies (Wang et al., 2020). Though most of the work have focussed on English, and not much work has been contributed when it comes to disfluencies in Indian Languages. This lack of research for Indian context can be attributed to the scarcity of labeled data and standardized annotation guidelines specific to Indian languages. In this research, we aim to bridge that very gap along with providing a robust algorithm for synthetic generation of required data.

An example sentence showcasing the importance of disfluency handling for MT (Hindi -> English):

तो उधर आपको सर ने क्या कहा था ? क्यों बोला था उनको ? बोला, क्या प्रॉब्लम है उनको ?

Corresponding Translation to English: So what did Sir tell you there? Why did you tell him? Said, What problem does he have?

Translation after removing disfluency: So what did Sir tell you there? What problem does he have?

The text in blue indicates the alteration, while the text in red indicates the reparandum (category: repair). After the reparandum is removed from the original text, the translation quality becomes better.

2. Related Work

Previous research has primarily focused on speech and spoken disfluency, with limited attention given to textual disfluencies. Moreover, research specifically addressing disfluencies in Indian languages is scarce. It is important to note that there is a notable absence of standardized annotation guidelines tailored for annotating disfluencies in Indian languages. Therefore, this is an aspect that we propose to address through our work.

A very efficient solution is available for generating disfluent data in English (Passali et al., 2022). However, when considering Indian languages, it is not feasible to directly apply similar algorithms for the reasons outlined above. (Bhat et al., 2023b) investigated a dataset for disfluency correction, though their focus was solely on Hindi among the Indian languages.

For Indian Languages, a zero shot detection of disfluencies along with synthetic data generation was shown to be very useful (Kundu et al., 2022). This shows us the reason why such synthetic augmentation can be so crucial. Due to the newly created dataset's wide range of variations and scenarios, it not only tackles the issues of data scarcity and class imbalance but also improves model generalisation. Additionally, research has demonstrated that adversarial training with actual data but a significant reliance on synthetic data also improves score. (Bhat et al., 2023a). Our analysis of disfluencies exhibits a finer granularity in annotation and construction, which extends to Indian languages and tries to surpass previous research efforts in this domain.

Disfluencies are perfectly natural, and do not sound wrong to the human ear. When we talk about disfluency correction, we primarily try to make the machine understand better. Ultimately, in the bigger picture of speech-to-speech machine translation, we would want the output to be as human-like as possible. Since disfluency in one language does not necessarily map one-to-one with another language, hence it is indispensable to know the complicacies regarding disfluencies in both source and target language.

The primary focus areas in this study are:

- Appropriate annotation guidelines that consider the subtleties of Indian languages
- Synthetic generation of such disfluencies using an algorithm that tries to improve on previous works
- Characteristics of Indian languages which might appear very similar, but are different to disfluencies
- How code-mixed data plays a role

Here we work on 6 Indian Languages namely: Hindi, Bengali, Marathi, Telugu, Kannada and Tamil.

3. Data

We used simulated conversations in authentic contexts for our investigation. This was obtained by us from the IIT Madras SPRING lab [2],- who had acquired this data from vendors on a payment basis followed by thorough quality check on the transcriptions. The dataset included both monologues and conversations between two to four persons.

Since monologues are usually prepared or practiced speeches, people frequently have the chance to plan and organise their speech beforehand, reducing the likelihood of disfluencies like pauses, hesitations, or self-corrections. Furthermore, the lack of instant input from listeners lessens the necessity for spontaneous alterations or changes during monologues. Thus we focussed on the natural conversational audios. We manually filtered

[2] https://asr.iitm.ac.in/dataset

Language	Synthetic	PMIndia	Real data
Hindi	7	1	5.5
Bengali	7	1	5.5
Marathi	7	1	2
Telugu	5	1	2.5
Kannada	7	1	3
Tamil	7	1	8

Table 1: Full distribution of data (in hours)

out those non-monologue audios which appeared to have good amount of disfluencies. We acquired the data for Hindi, Marathi, Bengali, Kannada and Tamil as mentioned above, whereas for Telugu, we collected the data using conversations from YouTube videos with creative commons licence. While annotating on the acquired transcripts, if any instance arose regarding incorrect transcription, we first rectify the transcript before proceeding with the annotation.
We also used approximately 1 hr[3] of data from the PMIndia (Haddow and Kirefu, 2020) corpus for each language, to which we synthetically added disfluency. This allowed to make our dataset more diverse.

The Table 1 shows the distribution and size of data set for each language(numbers).

3.1. Tagset Considered

The tags considered for annotating the data include:

- **Pet_r** : marks the reparandum under the category of pet_phrases
- **Filler_r** : marks the reparandum under the category of filler words/pauses.
- **Edit_r** : marks the edit terms, also called the interregnums (Kundu et al., 2022)
- **Repeat_r** : marks the reparandum under the category of repetition
- **Repair_r** : marks the reparandum under the category of repair
- **False_r** : marks the reparandum under the category of false start
- **Alteration**: marks the alteration where required.

[3]We approximate 6500-7000 words to be present in one hour of speech

3.2. Annotation Guidelines

In contrast to English, the datasets comprising six Indian languages lack comprehensive guidelines regarding their behavior concerning disfluency.
Hence we followed a holistic approach for identifying the instances which count as disfluency, along with identifying other minute details which need to be given special attention to while dealing with Indian languages.
A pivotal concept reiterated throughout is the variability of words or phrases that may exhibit disfluency in certain contexts but not in others. This variability hinges on whether the word or phrase carries semantic significance in the given context.

The following examples use red text to indicate reparandum, green text to indicate editing terms, and blue text to indicate alteration. Unless specifically mentioned, the non-English examples are in Hindi. For all the non-English text, its corresponding transliteration is present under the respective texts.

3.2.1. Filled Pauses/Filler Words

This category encompass the phenomena when speakers tend to use certain sounds like 'uh', 'uhmm' in between their utterances. These do not carry any meaning, and in most cases just a sign of the speaker thinking and speaking simultaneously. Important exception: the cases of interjections and discourse markers. There are cases where certain filler words are used as meaningful interjections/discourse markers. In such cases, they should not be marked as disfluency.

Examples:

- Hindi: मैं अ कल तक अ पहुंच जाउँगा
  mai uh kal tak uh pohoch jaunga
- Tamil: அவளுக்கு ஒரு ம்ம் குறுஞ்செய்தி அனுப்பு
  Avaḷukku oru m'm kurunceyti anuppu

3.2.2. Pet Phrases

Many speakers use particular terms rather frequently, even in situations where their semantic contribution is negligible. These terms are called "pet phrases", and they can include discourse markers and common interjections. Moreover, these catchphrases are unique to each speaker, and there is no set list of terms that they can use as their pet phrases.

Examples:

- Hindi: मतलब यह बात मतलब एक मेम बोले थे
  matlab yah baat matlab ek ma'am bole the

- Marathi: आपन काल ते खाल्लं नं, ते आपलं खरबूज, ते खूप छान होतं.
  aapan kaal te khaalla na, te aapla kharbuj, te khup chaan hota

3.2.3. Repetitions

These are the simple cases when speaker repeats certain words/phrases in continuation. We need to be cautious when dealing with the concept of reduplication and emphasis in Indian Languages (Section 3.3). Those cases should not be marked as disfluency.

Examples:

- Hindi: एक नार्मल डॉक्टर का डॉक्टर का उस दिन आना जरूरत था ।
  ek normal doctor ka doctor ka us din aana jaroorat tha

- Bengali: আমি বাড়িতে পৌঁছে এটি সমাধান এটি সমাধান করার চেষ্টা করব
  aami barite paunche eti samaadhaana eti samaadhaana korar chesta korbo

3.2.4. Repair

There are many instances where the speaker utters words and phrases, but then realizes his mistake, and corrects it. The part which he uttered by mistake is part of the reparandum, and the alteration contains the part to be replaced with. Importantly, the topic remains the same. There are cases of emphasis, code mixing, echo words, abrupt endings, phrase insertion(gaps) which should not be confused with the phenomena of repair. Section 3.3 contains details for all such cases.

Examples:

- Hindi: वी थिंक दॅट मतलब हम बस एक ही चीज सोचते कि इन्हें बेस्ट पॉसिबल ट्रीटमेंट मिले, चाहे वो कैसे भी मिले.
  we think that matlab hum bas ek hi cheez sochte ki inhe best possible treatment mile, chahe wo kaise bhi mile

- Telugu: నేను రేపు అహ ఎల్లుండి వెళ్తున్న,
  nenu repu aha ellundi veltunna

- Bengali: আমার ফ্লাইট আগামীকাল সকাল ৭টায় বিকাল ৭টায়.
  aamaar flight aagamikaal shokal shaattaay bikal shaattaay

3.2.5. False Start

In this phenomena, the speaker abandons his utterance midway through and starts with another utterance with a different topic. We simply note the editing and reparandum terms since false starts indicate that the speaker is beginning over. This is due to the lack of clarity regarding the precise alteration that would be made — either the entire sentence or just a portion of it. As a result, we do not mark any alterations to avoid any ambiguity.

Examples:

- Hindi: मैंने अ.. ऑलरेडी मेरा इन्शुरन्स इनिशिएट हो चुका था ।
  maine uhh.. already mera insurance initiate ho chuka tha

- Marathi: कालचा एपिसोड तर... अरे आपल्याला दिवाळी च्या सुट्ट्या कधी आहेत ?
  kaalcha episode tar... arey aaplyala diwali chya suttya kadhi aahet?

3.2.6. Edit term

These are the lexical cues which indicate the end of reparandum and start of alteration. It can be filler words/pet phrases, or some words distinctively carrying the meaning of 'apology the unintended utterance(reparandum)' which are exclusively considered as edit terms like "sorry", "i mean" in English. These are marked in the above mentioned examples in green color.

3.3. Corner cases while annotating disfluencies

All the below instances are not to be considered as disfluencies, except code mixing in certain scenarios, as explained.

3.3.1. Code Mixing

Many instances involving code mixing, may or may not be part of disfluencies. This can be identified as following.

Cases when Code Mixing is NOT disfluency:

- **Simple Code mixing**: This is when we replace the words/phrases of one language with another, without interrupting the flow of speech.

  Example: I was going to reach my home कि माँ का फोन आ गया
  I was going to reach my home ki maa ka phone aa gaya

- **Emphasis**: There will be instances where a speaker deliberately utters a sentence in one language and repeats in another, to emphasise

its importance. Usually this kind of emphasis occurs when the whole clause/sentence is repeated in another language. The most helpful cue to detect emphasis of such kind is from the audio.

Example: I will do the work by tomorrow मैं कल तक काम कर लूंगा ।
I will do the work by tomorrow main kal tak kaam kar lunga

- **Situational Code mix**: These are instances when a speaker deliberately uses another language and repeats what he said, to make sure the other speaker is following him.

Cases when Code Mixing leads to Disfluency:

- When the speaker starts in one language, abandons it midway and utters the same sentence in another language. - this will be an instance of *repair* type of disfluency

  Example: आइ विल मैं कल तक काम कर लूंगा
  I will main kal tak kaam kar lunga

- When the speaker starts in one language, abandons it and utters a new sentence with topic change in another language - this will be an instance of *false start* type of disfluency.

  Example: Her name मैं वहाँ दस बजे तक पहुँच जाऊँगा
  her name main wahan das baje tak pahunch jaunga

Both these techniques were also applied while generating synthetic disfluencies.

3.3.2. Reduplication

This is a phenomenon widely present in Indian Languages. The speaker deliberately repeats certain words to convey some meaning/emphasize.

Example: ऐसे छोटे छोटे बातों पे वो चिल्लाने लगते थे ।
aese chote chote baaton pe wo chillane lagte the

3.3.3. Echo Words

This is a similar phenomenon to reduplication, but the words are not exactly copied, rather they sound/rhyme similar.

Example: मतलब सिस्टर–विस्टर से पूछने की कोशिश करते हैं
matlab sister-wister se puchne ki koshish karte hain

3.3.4. Emphasis

- **By Repetition**: Frequently, speakers intentionally repeat a word or phrase to emphasize a point or convey specific meaning.

  Example: तो जनरल वार्ड में पहले मेडिसिन्स ही चल रहा था । चल रहा था ,चल रहा था ।
  toh general ward me pehle medicines hi chal raha tha chal raha tha chal raha tha.

  In this instance, despite the repetition of words or phrases, nothing has been labeled as reparandum or alteration. This repetition is intentional on the part of the speaker telling about the continuous process of administering medicines.

- **Numbers**: Speakers often emphasise on what they want to convey by simply repeating the numbers or say the same thing in different languages(code mixed). Such cases are not to be considered disfluency.

  Example: बहोत कॉस्टली है, पैंतीस से चालीस हजार थर्टी फाईव्ह टू फोर्टी थाउजंड पर डे, लग रहे है
  bohot costly hai, pentees se chaalees hazaar thiry five to forty thousand per day lag rahe hain

  In this example, there would not be any disfluency - neither repair nor a code mixed repeat. By repeating in different languages, the speaker simply emphasizes the huge sum the figure represents.

- **Specificity**: Speakers often tend to specify about what they uttered, giving specificity to certain words/nouns.

  Example: मुझे क्रिकेट खलेने के लिए बॉल लाल बॉल चाहिये.
  mujhe cricket khelne ke liye ball laal ball chahiye

3.3.5. Abrupt Endings

There is also the presence of 'abrupt endings', where the speaker altogether leaves some useful meaningful utterance midway and starts other utterance. In such cases, they are not disfluency.

Consider the text:
तो उसे अगर इधर दर्द देता है तो सिस्टर को बुला कर थोड़ा मतलब ये प्रॉब्लम नहीं है । तो वो सिस्टर्स ध्यान से देख लेते हैं।
**toh use agar idhar dard deta hai to sister ko bula kar thoda matlab* ye problem nahi hai. toh wo sisters dhyan se dekh lete hai .*

Here, following the portion enclosed in asterisks, the speaker discontinues and initiates a fresh expression or, alternatively, substitutes the entire phrase with 'ये'. One cannot consider this as disfluency since all parts of the asterisked utterance is important and conveys some information. Removing them would result in loss of information - which is not what disfluency represents.

3.3.6. Different Speakers

It is essential to note that when marking any utterance as disfluent, we must ensure that the suspected disfluent utterances originate from a single speaker. Thus the cases of *'echoic utterance'* or *'echoic questioning'*, indicated by a speaker repeating or echoing the listener's response immediately - should not be termed as disfluency.

Likewise, there are scenarios in which a speaker is interrupted mid-sentence by another speaker, resulting in an apparent disruption in the conversation flow. Nevertheless, such instances should not be labeled as disfluencies.

Addressing these nuances poses a significant challenge in disfluency identification tasks, particularly in the context of real conversations.

3.3.7. Phrase Insertion or Gaps

Indeed, it is common during speech for individuals to interject additional information abruptly to provide better context before resuming their original train of thought. Such instances should not be termed as disfluencies.

Consider the following example:
तो उसके बाद वो बोले मेरे से बात हुआ था कि ऐसा ऐसा है, तो आपको इमिडिएटली पैसा रिलिज करना पडेगा ।
**toh uske baad wo bole* mere se baat hua tha ki esa esa hai to aapko immediately paisa release karna padega*

If we observe, the portion enclosed in asterisks is where the speaker moved off from his speech, got some additional information as underlined, and then continued from where he left off. Thus these types of instances are special to spoken language, but not any disfluency.

Figure 1 shows a glimpse of the in-house developed tool to perform the annotations. We used our own tool so that we can easily customize the tags as well as carry out simultaneous editing of subtitles with respect to the audio playback.

4. Inter Annotator Agreement

Two annotators worked on each language for the task of annotating disfluencies. Thus correspondingly, the inter annotator agreement was done on 1 hr of data of each language. To calculate the IAA, Cohen's kappa coefficient was used - giving a score of 82-92% for the disfluency annotation on each language. Table 2 shows the kappa scores for the annotations done for disfluencies on Indian Languages.

Upon thorough analysis, it was found that pet phrases and filler words had the highest degree of agreement among annotators. On the other hand, there were some differences in the repair and false start annotation cases. It is crucial to remember that these differences did not always imply different annotations for the reparandum as a whole. Rather, disagreements primarily centered around the span of words marked for the reparandum and alteration. Considering the simplicity of filler words compared to the complexity and intricacies of repair annotations, this result is in line with our predictions.

Language	Score
Hindi	92
Bengali	89
Marathi	83
Telugu	86
Kannada	82
Tamil	85

Table 2: Kappa metric scores for annotation in each language

Thus in this task of disfluency identification, about 85-86% of inter-annotator agreement on average was reached. This level of agreement shows that we have a strong and dependable method for spotting and categorizing speech interruptions in different language situations.

5. Synthetic Data

When synthetically augmenting data with disfluencies, we have to keep in mind to make the disfluent data look as natural as possible.
To achieve the same, we synthetically generated the five categories along with paying attention to the different kinds possible within each category.

5.1. Filler words, Pet Phrases

For each language we had collected a list of commonly used filler words. Then for the given sentence, a random position is chosen, except the last position, and a random filler word from the list is concatenated at that position.
It is significant to note that when a speaker utters

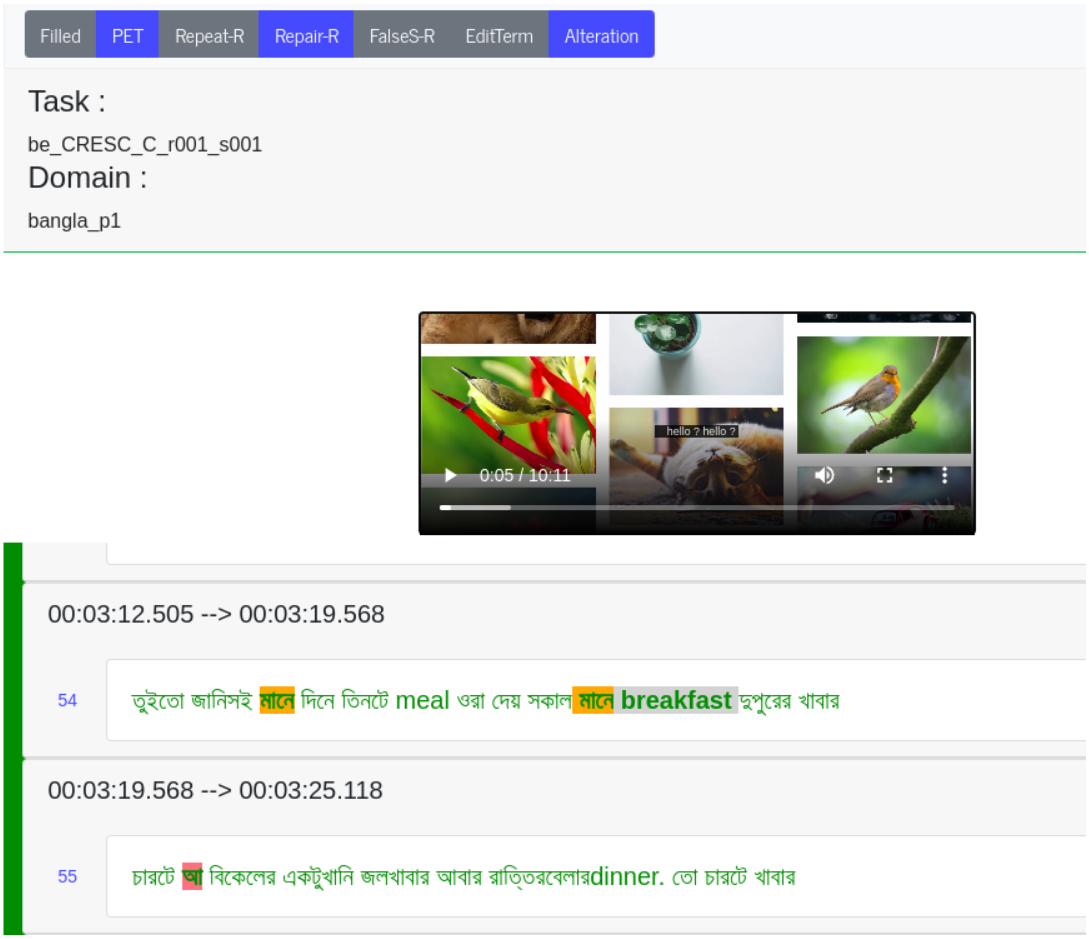


Figure 1: Tool used for Annotation(in-house developed tool). This is just an illustrative image showing the audio playback along with the corresponding subtitles (here in Bengali). The annotator can select the text and click on the appropriate category from the tabs present at the top. Each category will have its unique highlight color as well. The categories are namely: Filled Pause, Pet Phrase, reparandums of Repair, Repeat and False Start, Edit Terms and Alteration.

a filler word, he does not do it just once. Since fillers are a sign that the speaker is thinking and speaking, filler words usually occur more than once in an utterance or a sentence.
Hence following the same notion, if we are injecting a filler word to a sentence, we take into the account the length of the sentence, a probability measure 'p', and the max filler words that a sentence can accommodate - which we capped at 4. Thus using this methodology, we augmented filler words in a given sentence.
What distinguishes a pet phrase from a filler word in this methodology is the notion that pet phrases are unique to an individual. Therefore, if the speaker's identity is known in advance, the same pet phrase previously used by that individual is employed with higher likelihood.

5.2. Repetition

To add Repetition type of disfluency, we follow a similar approach as mentioned in (Kundu et al., 2022).

- **Word Repetition** : To implement this we choose a word randomly and repeat it.
- **Phrase Repetition** : We repeat an n-gram of two to five words. We initially use a weighted distribution of [0.4, 0.3, 0.2, 0.1] to randomly select a length from [2, 3, 4, 5].

5.3. Repair

The phenomenon of repair has many nuances if we observe carefully.

- **Partial Word**: This represents the concept wherein a speaker partially utters a word, then properly utters it afterwards - closely related to stammering.

  Attention was paid on how and where such disfluencies occur. We came up with the idea that there is a very low probability of having

a partial word type of disfluency if the actual word has less than 7 unicode characters.

Among those words which have more than 7 characters, one of them is chosen at random. For that chosen word, a random position is chosen till which the partial word would be created. Thus the partial word formed is our reparandum and is placed before the original word.

- **Phrase Repair**:

  This encircles the typical case of repair disfluency, also called correction. First, we randomly choose 2-6 contiguous words. Then we apply the idea that in a typical correction, the lexical item/POS tag or some related feature of either the first or last word remains the same in reparandum and alteration. Thus keeping either the first or last word unchanged, we modify the rest of the words. To achieve this task, we use Muril (Khanuja et al., 2021) and applied fill mask algorithm sequentially - to get as natural sounding text as possible with respect to the new words that are being generated.

- **Code Mix Repair**:

  Code mix disfluencies is the area where not much has been thought into in the field. We tried to replicate the behaviour of code mix disfluency taking the help of LTRC translation engine [4] .

  Unlike phrase repair where we could simply replace the tokens using fill mask, here we cannot simply replace the tokens with their translations. The main reason being the grammar and sentence structure of the languages involved along with other factors of translation. Hence we translate the whole sentence, and then randomly first k words. This acts as the reparandum, alteration being the full sentence. Note that we only dealt with *Indian Language - English* code mixed data.

5.4. False Start

First, we choose two distinct sentences at random to produce false starts. Subsequently, we divide the initial sentence into two parts at random and join the first segment of the split with the second sentence. With a random probability, instead of simply splitting the sentence, we first translate it, then split and follow the same method, to produce a more natural sounding code mixed false start.

Algorithm 1 shows the entire algorithm for synthetically generating the disfluencies.

[4] https://ssmt.iiit.ac.in/translate

6. Experiment and Methodology

For this task, we used XLM-RoBERTa (Conneau et al., 2019) which is a multilingual version of RoBERTa (Liu et al., 2019). Having about 125M parameters, it is trained on on 2.5 TB of filtered CommonCrawl data in 100 languages with the Masked language modeling (MLM) objective.

We fine-tuned the XLM-RoBERTa model for classification task on our training dataset - which comprises synthetic data as well as real-world annotated data. Table 3 shows the hyperparameters used for the fine-tuning task.

Parameter	Value
Optimizer	Adam
LR	3e-5
Wt Decay	0.01
Batch Size	16
Epochs	10

Table 3: Hyperparameters used for the fine-tuning of XLM-RoBERTa for disfluency identification.

We set the *P_disf* in the Algorithm 1 such that the overall disfluency percentage(by words) in the data stays in the range 8-10% so as to mimic real world data.

7. Performance & Observation

We fine-tuned the model on the generated synthetic data along with real-world annotated data. An additional collection of annotated data from the real world was used for testing. Languages like Hindi and Tamil gave decent results of F1 scores >35. One reason their ratings exceed those of other languages could be attributed to a higher availability of data. Marathi and Telugu gave the lowest of scores, primarily because of less amount of real world data available for them.

Another experiment was run, this time by changing the synthetic augmentation algorithm to produce a better distribution of all the disfluency cate-

Language	Test data (in hrs)	Exp. 1	Exp. 2
Hindi	2	59	60
Bengali	2	22	25
Marathi	1	8	9
Telugu	1	9	9
Kannada	1	16	20
Tamil	2	35	43

Table 4: Weighted F1 scores for the task of disfluency identification.

Algorithm 1 Synthetically augmenting disfluencies

Require:
Sentence or a text on which disfluency needs to be added.
list of filler words *FW*, pet phrases *PP* and edit terms *ET* in each language.
Ensure: text filled with disfluency.
P_disf = A probability which decides whether disfluency will be injected in current sentence or not
if P_disf == True **then**
    D_{type} = randomly choose the type of disfluency to inject
    if D_{type} in (filler words, pet phrase) **then**
        P_{fw} = probability to inject multiple filler words
        P_{pp} = probability to inject multiple pet phrase
        pos = random position to inject filler word/pet phrase
        generate_disfluency($text$, pos, P_{fw}, P_{pp}, FW, PP) ▷ This adds the disfluent words at the position and returns the final synthesized text
    else if D_{type} in (repeat, repair, false start) **then**
        $start\_pos$, end_pos = randomly choose the starting position(word) and the end position.
        $rep\_substring$ = text[$start\_pos$: end_pos] ▷ this substring acts as the alteration
        add_edit_terms($text$, end_pos, FW, PP, ET) ▷ This internally adds edit terms/filler words/pet phrases or a combination of them to the end of the chosen substring
        generate_disfluency($text$, $start\_pos$, end_pos, $rep_substring$) ▷ This adds the generated reparandum before the chosen substring and returns the details of reparandum, alteration and the final synthesized text
    end if
end if

gories. The *P_disf* in the Algorithm 1 was tweaked such that the overall disfluency percentage was around 21%. Additionally, improving the distributions among the various disfluency categories was an important point that we worked on. To get better distribution and more quantity of repairs, the whole list of categories and subcategories of disfluencies were flattened into one list. This ensured that each kind of disfluency (sub)category will have equal probability. The Table 4 shows the weighted F1 scores calculated for both the experiments.

8. Conclusion and Future Work

We have presented detailed guidelines for annotating disfluencies in real-world conversations, accompanied by an algorithm for synthesizing such disfluencies in the data. The necessity of this thorough annotation is underscored by the complexity of the task, as evidenced by the obtained scores, indicating that identifying disfluencies for Indian Languages in continuous real-world conversations poses a significant challenge. Furthermore, the synthetic augmentation process requires constant refinement to better emulate real-world disfluencies. The guidelines outlined in Section 3.3 highlight numerous subtle nuances that models must learn to accurately identify or not identify them as disfluencies.

In our approach, we started with the acquired transcripts and progressed from there. Therefore, it stands to reason that a higher-quality ASR system with as low Word Error Rate (WER) as possible would enhance the efficacy of the entire workflow.

Going forward, there are several directions in this field that has to be explored. It is imperative to continuously refine the annotation criteria in addition to gathering additional datasets, especially from real-world sources. Furthermore, it is expected that some intelligent usage of semantic knowledge pertaining to punctuations, grammatical chunks or part-of-speech tags will improve the algorithm's overall performance. These might have a crucial role both while artificially synthesizing disfluent data as well as for disfluency identification task.

References

Vineet Bhat, Preethi Jyothi, and Pushpak Bhattacharyya. 2023a. Adversarial training for low-resource disfluency correction. *arXiv preprint arXiv:2306.06384*.

Vineet Bhat, Preethi Jyothi, and Pushpak Bhattacharyya. 2023b. Disco: A large scale human annotated corpus for disfluency correction in indo-european languages. *arXiv preprint arXiv:2310.16749*.

Marcus Colman and Patrick Healey. 2011. The distribution of repair in dialogue. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33.

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. *arXiv preprint arXiv:1911.02116*.

Aditya Gupta, Jiacheng Xu, Shyam Upadhyay, Diyi Yang, and Manaal Faruqui. 2021. Disfl-qa: A benchmark dataset for understanding disfluencies in question answering. *arXiv preprint arXiv:2106.04016*.

Barry Haddow and Faheem Kirefu. 2020. Pmindia–a collection of parallel corpora of languages of india. *arXiv preprint arXiv:2001.09907*.

Peter A Heeman and James Allen. 1999. Speech repains, intonational phrases, and discourse markers: modeling speakers' utterances in spoken dialogue. *Computational Linguistics*, 25(4):527–572.

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*.

Rohit Kundu, Preethi Jyothi, and Pushpak Bhattacharyya. 2022. Zero-shot disfluency detection for indian languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4442–4454.

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*.

Tatiana Passali, Thanassis Mavropoulos, Grigorios Tsoumakas, Georgios Meditskos, and Stefanos Vrochidis. 2022. Lard: Large-scale artificial disfluency generation. *arXiv preprint arXiv:2201.05041*.

Sharath Rao, Ian Lane, and Tanja Schultz. 2007. Improving spoken language translation by automatic disfluency removal: Evidence from conversational speech transcripts. In *Proceedings of Machine Translation Summit XI: Papers*.

Elizabeth Ellen Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis, Citeseer.

Shaolei Wang, Wangxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang. 2020. Multi-task self-supervised learning for disfluency detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9193–9200.

Wen Wang, Gokhan Tur, Jing Zheng, and Necip Fazil Ayan. 2010. Automatic disfluency removal for improving spoken language translation. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5214–5217. IEEE.

EmoMix-3L: A Code-Mixed Dataset for Bangla-English-Hindi Emotion Detection

Nishat Raihan*, Dhiman Goswami*, Antara Mahmud, Antonios Anastasopoulos, Marcos Zampieri

George Mason University, USA
{mraihan2, dgoswam}@gmu.edu

Abstract

Code-mixing is a well-studied linguistic phenomenon that occurs when two or more languages are mixed in text or speech. Several studies have been conducted on building datasets and performing downstream NLP tasks on code-mixed data. Although it is not uncommon to observe code-mixing of three or more languages, most available datasets in this domain contain code-mixed data from only two languages. In this paper, we introduce EmoMix-3L, a novel multi-label emotion detection dataset containing code-mixed data from three different languages. We experiment with several models on EmoMix-3L and we report that MuRIL outperforms other models on this dataset.

Keywords: Code Mixing, Dataset, Emotion Detection

1. Introduction

The ability to convey emotions is an essential part of human communication. NLP models have been applied to detect emotions (e.g., anger, fear, joy) in texts from social media (Gaind et al., 2019), customer service (Gupta et al., 2010), and healthcare (Ayata et al., 2020). Emotion detection is an important part of social media analysis and mining efforts that include popular tasks such as sentiment analysis (Liu, 2020) and stance detection (Kawintiranon and Singh, 2021).

Most studies on sentiment analysis and emotion detection are carried out in one language at a time (Abdul-Mageed and Ungar, 2017; Chatterjee et al., 2019). Apart from a few notable exceptions (Vedula et al., 2023), detecting emotion in multilingual and code-mixed environments has not been significantly explored. Code-mixing is very common in multilingual societies. It is defined as the practice of using words and grammatical constructions from two or more languages interchangeably (Muysken, 2000). Code-mixing can occur at various levels such as intra-sentential where code-mixing is present within a sentence, and inter-sententialwhere code-mixing is present across sentences.

Detecting sentiments and emotions in code-mixed texts is a challenging task that we address in this paper by introducing EmoMix-3L, a multi-label emotion detection containing Bangla, Hindi, and English code-mixed texts. These three languages are often used together by the population of West Bengal. They are also used by populations from South East Asian living in other parts of the world where English is spoken as the official language or *lingua franca* such London, New York, or Singapore. Recent studies have created resources for these three languages in tasks such as sentiment analysis and offensive language detection (Raihan et al., 2023a; Goswami et al., 2023). To the best of our knowledge, however, no datasets for emotion detection in Bangla-English-Hindi code-mix exists and EmoMix-3L fills this gap.

The main contributions of this paper are as follows:

- We introduce EmoMix-3L[1], a novel three-language code-mixed test dataset in Bangla-Hindi-English for multi-label emotion detection. EmoMix-3L contains 1,071 instances annotated by speakers of the three languages. We make EmoMix-3L freely available to the community.

- We provide a comprehensive evaluation of several monolingual, bilingual, and multilingual models on EmoMix-3L.

We present EmoMix-3L exclusively as a test set due to the unique and specialized nature of the task. The size of the dataset, while limited for training purposes, offers a high-quality testing benchmark with gold-standard labels. Given the scarcity of similar datasets and the challenges associated with data collection, EmoMix-3L provides an important resource for the evaluation of multi-label emotion detection models, filling a critical gap in multi-level code-mixing research.

[1] https://github.com/GoswamiDhiman/EmoMix-3L

WILDRE-7 Workshop, pages 11–16

2. Related Work

A few studies studies have addressed emotion detection on bilingual code-mixed data (Wadhawan and Aggarwal, 2021; Vedula et al., 2023; Ameer et al., 2022). Vedula et al. (2023) implemented a multi-class emotion detection model leveraging transformer-based multilingual Large Language Models (LLMs) for English-Urdu code-mixed text. However, the study's ability to interpret code-mixed sentences that combine English and Roman Urdu had limitations. The study by Ameer et al. (2022) highlights how multi-label emotion classification may be used to identify every emotion that could exist in a given text. 11,914 code-mixed (English and Roman Urdu) SMS messages make up the substantial benchmark corpus presented in this paper for the multi-label emotion classification challenge.

There have been a number studies on Bengali-English code-mixed data. Mursalin et al. (2022) used deep learning approaches to identifying emotions from texts containing mixed Bengali and English coding, with an emphasis on comparing and contrasting the effectiveness of the suggested model with other ML and DL methods already in use. Ahmad et al. (2019) have explored and analyzed regional Indian code-mixed data. In this paper, the importance and applications of sentiment detection in a variety of domains are discussed, with an emphasis on Indo-Aryan languages like Tamil, Bengali, and Hindi.

A few studies have addressing Bengali-English-Hindi code-mixing on social media. Raihan et al. (2023a) uses multiple monolingual, bilingual, and multilingual models and a unique dataset with gold standard labels for sentiment analysis in Bangla-English-Hindi. Goswami et al. (2023) presents a novel offensive language identification dataset with the same three languages. Finally, another similar work include the TB-OLID dataset (Raihan et al., 2023b) that contains both transliterated and code-mixed data for offensive language identification.

3. The *EmoMix-3L* Dataset

We choose a controlled data collection method, asking the volunteers to freely contribute data in Bangla, English, and Hindi. This decision stems from several challenges of extracting such specific code-mixed data from social media and other online platforms. Our approach ensures data quality and sidesteps the ethical concerns associated with using publicly available online data. Such types of datasets are often used when it is difficult to mine them from existing corpora. As examples, for fine-tuning LLMs on instructions and conversations, semi-natural datasets like Databricks (2023) and Nie (2023) have become popular.

Data Collection Ten undergraduate students fluent in the three languages was asked to prepare 300 to 350 social media posts each. They were allowed to use any language, including Bangla, English, and Hindi to prepare posts on several daily topics like politics, sports, education, social media rumors, etc. We also ask them to switch languages if and wherever they feel comfortable doing so. The inclusion of emojis, hashtags, and transliteration was also encouraged. The students had the flexibility to prepare the data as naturally as possible. Upon completion of this stage, we gathered 2,598 samples that contained at least one word or sub-word from each of the three languages using langdetect (Mazzocchi, 2012) an open-sourced Python tool for language identification.

Data Annotation We annotate the dataset in two steps. Firstly, we recruited three students from social science, computer science, and linguistics fluent in the three languages to serve as annotators. They annotated all 2,598 samples with one of the five labels (Happy, Surprise, Neutral, Sad, Angry) with a raw agreement of 47.9%. We then take 1,246 instances, where all three annotators agree on the labels, and use them in a second step. To further ensure high-quality annotation, we recruit a second group of annotators consisting of two NLP researchers fluent in the three languages. After their annotation, we calculate a raw agreement of 86% (Kvålseth, 1989), a Cohen Kappa score of 0.72. After the two stages, we only keep the instances where both annotators agree, and we end up with a total of 1,071 instances. The label distribution is shown in Table 1.

Label	Instances	Percentage
Happy	228	21.29%
Surprise	227	21.20%
Neutral	223	20.82%
Sad	205	19.14%
Angry	188	17.55%
Total	1,071	100%

Table 1: Label distribution in EmoMix-3L

Dataset Statistics A detailed description of the dataset statistics is provided in Table 2. Since the dataset was generated by people whose first language is Bangla, we observe that the majority of tokens in the dataset are in Bangla. There are several *Other* tokens in the dataset that are not from Bangla, English, or Hindi language. The *Other* tokens in the dataset primarily contain transliterated words as well as emojis and hashtags. Also, there are several misspelled words that have been classified as *Other* tokens too.

	All	Bangla	English	Hindi	Other
Tokens	98,011	36,784	6,587	15,560	39,080
Types	21,766	9,118	1,237	1,523	10,022
Avg	91.51	34.35	6.15	14.53	36.49
Std Dev	20.24	9.13	2.88	5.94	10.64

Table 2: EmoMix-3L Data Card. The row *Avg* represents the average number of tokens with its standard deviation in row *Std Dev*.

Synthetic Train and Development Set We present EmoMix-3L as a test dataset and we build a synthetic train and development set that contains Code-mixing for Bangla, English, and Hindi. We use an English training dataset annotated with the same labels as EmoMix-3L, namely Social Media Emotion Dataset (SMED)[2]. We then use the *Random Code-mixing Algorithm* (Krishnan et al., 2022) and *r-CM* (Santy et al., 2021) to generate the synthetic Code-mixed dataset. Similar approach is also found in (Gautam et al., 2021).

4. Experiments

Monolingual Models We use six monolingual models for these experiments, five general models, and one task fined-tuned model. The five monolingual models are DistilBERT (Sanh et al., 2019), BERT (Devlin et al., 2019), BanglaBERT (Kowsher et al., 2022), roBERTa (Liu et al., 2019), HindiBERT (Nick Doiron, 2023). BanglaBERT is trained in only Bangla and HindiBERT in only Hindi while DistilBERT, BERT, and roBERTa are trained in English only. Finally, the English task fine-tuned model we use is emoBERTa (Kim and Vossen, 2021).

Bilingual Models BanglishBERT (Bhattacharjee et al., 2022) and HingBERT (Nayak and Joshi, 2022) are used as bilingual models as they are trained on both Bangla-English and Hindi-English respectively.

Multilingual Models We use mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) as multilingual models which are respectively trained on 104 and 100 languages including Bangla-English-Hindi. We also use IndicBERT (Kakwani et al., 2020) and MuRIL (Khanuja et al., 2021) which cover 12 and 17 Indian languages, respectively, including Bangla-English-Hindi. We also perform hyper-parameter tuning while using all the models to prevent overfitting.

Prompting We use prompting with GPT-3.5-turbo model (OpenAI, 2023) from OpenAI for this task. We use the API for zero-shot prompting (see Figure 1) and ask the model to label the test set.

Additionally, we run the same experiments separately on synthetic and natural datasets splitting both in a 60-20-20 way for training, evaluating, and testing purposes.

Role: "You are a helpful AI assistant. You are given the task of Emotion Detection.

Definition: An emotion is a feeling that can be caused by the situation that people are in or the people they are with. You will be given a text to label either 'Happy', 'Surprise', 'Neutral', 'Sad' or 'Angry'.

Task: Generate the label for this **"text"** in the following format: <*label*> *Your_Predicted_Label* <*label*>. Thanks."

Figure 1: Sample GPT-3.5 prompt.

5. Results

In this experiment, synthetic data is used as a training set, and natural data is used as the test set. The F1 scores of monolingual models range from 0.14 to 0.41, where roBERTa performs the best. MuRIL is the best of all the multilingual models, with an F1 score of 0.54. Besides, a zero-shot prompting technique on GPT 3.5 turbo provides a 0.51 weighted F1 score. The task fine-tuned model emoBERTa provides the F1 score of 0.42. BanglishBERT scores 0.44 which is the best F1 score among all the bilingual models. These results are available in Table 3.

Models	F1 Score
MuRIL	**0.54**
XLM-R	0.51
GPT 3.5 Turbo	0.51
BanglishBERT	0.44
HingBERT	0.43
emoBERTa	0.42
roBERTa	0.41
BERT	0.38
mBERT	0.35
DistilBERT	0.24
IndicBERT	0.22
BanglaBERT	0.16
HindiBERT	0.14

Table 3: Weighted F-1 score for different models: training on synthetic and tested on natural data (EmoMix-3L).

We perform the same experiment using synthetic data for training and testing. We present results in Table 4. Here, MuRIL with 0.67 F1 score is the

[2] https://www.kaggle.com/datasets/gangulyamrita/social-media-emotion-dataset

best-performing model. BERT is the best among the monolingual models where their F1 range from 0.32 to 0.45. BanglishBERT with 0.47 F1 score is the best among the bilingual models. The task fine-tuned model emoBERTa scores 0.41 for the synthetic dataset.

Models	Weighted F1 Score
MuRIL	**0.67**
XLM-R	0.51
mBERT	0.49
BanglishBERT	0.47
HingBERT	0.45
BERT	0.44
emoBERTa	0.41
roBERTa	0.41
DistilBERT	0.40
BanglaBERT	0.39
IndicBERT	0.38
HindiBERT	0.32

Table 4: Weighted F-1 score for different models: training on synthetic and tested on synthetic data.

5.1. Error Analysis

The confusion matrix for the best-performing model MuRIL for training on synthetic and tested in EmoMix-3L is shown in Figure 2.

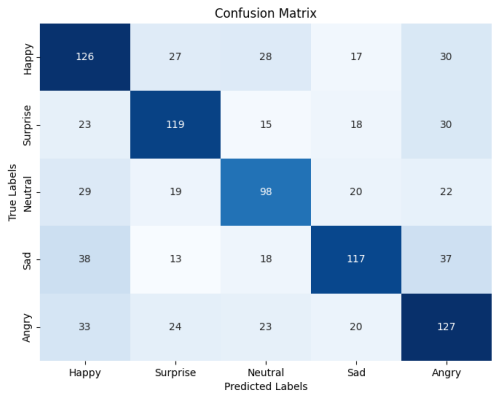


Figure 2: Confusion Matrix (Training on synthetic data, tested on EmoMix-3L).

We observe *Other* tokens in more than 39% of the whole dataset, as shown in Table 2. These tokens occur due to transliteration which poses a challenge for most of the models since not all of the models are pre-trained on transliterated tokens. BanglishBERT did well since it recognizes both Bangla and English tokens. However, the total number of tokens for Hindi-English is less than Bangla-English tokens, justifying HingBERT's inferior performance compared to BanglishBERT (see Table 3). Also, misspelled words and typos are also observed in the datasets, which are, for the most part, unknown tokens for the models, making the task even more difficult. Some examples are available in Appendix A which are classified wrongly by all the models.

6. Conclusion and Future Work

We introduce EmoMix-3L, a novel dataset containing 1,071 instances of Bangla-English-Hindi code-mixed content. We have also created 100,000 instances of synthetic data in the same languages to facilitate our training methods. We have tested multiple monolingual models on these datasets, and MuRIL performs the best, especially when it was trained on synthetic data and tested on EmoMix-3L. MuRIL was also the best in the scenario of both training and testing on synthetic data, outperforming all the other models in multi-label emotion detection. Looking ahead, we would like turning EmoMix-3L into a larger dataset serving both as a training and testing dataset. We would also like to create pre-trained tri-lingual code-mixing models. It will facilitate the emotion detection task in the intricate mix of Bangla, English, and Hindi. Moreover, we would like to explore the performance of large language models by fine-tuning them on code-mixed datasets. This will provide valuable insights into their unexplored training corpora and their ability to cope with code-mixed scenarios.

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of ACL*.

Gazi Imtiyaz Ahmad, Jimmy Singla, and Nikita Nikita. 2019. Review on sentiment analysis of indian languages with a special focus on code mixed indian languages. In *Proceedings of ICACTM*.

Iqra Ameer, Grigori Sidorov, Helena Gomez-Adorno, and Rao Muhammad Adeel Nawab. 2022. Multi-label emotion classification on code-mixed text: Data and methods. *IEEE Access*, 10:8779–8789.

Değer Ayata, Yusuf Yaslan, and Mustafa E Kamasak. 2020. Emotion recognition from multimodal physiological signals for emotion aware healthcare systems. *Journal of Medical and Biological Engineering*, 40:149–157.

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 ACL*.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. Semeval-2019 task 3: Emocontext contextual emotion detection in text. In *Proceedings of SemEval*.

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*.

Databricks. 2023. Dolly 2.0: An open source, instruction-following large language model. Accessed: 2023-09-10.

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*.

Bharat Gaind, Varun Syal, and Sneha Padgalwar. 2019. Emotion detection and analysis on social media. *arXiv preprint arXiv:1901.08458*.

Devansh Gautam, Prashant Kodali, Kshitij Gupta, Anmol Goel, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. Comet: Towards code-mixed translation using parallel monolingual sentences. In *Proceedings of CALCS*.

Dhiman Goswami, Md Nishat Raihan, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. OffMix-3L: A novel code-mixed test dataset in bangla-english-hindi for offensive language identification. In *Proceedings of SocialNLP*.

Narendra Gupta, Mazin Gilbert, and Giuseppe Di Fabbrizio. 2010. Emotion detection in email customer care. In *Proceedings of NAACL*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik 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 the ACL*.

Kornraphop Kawintiranon and Lisa Singh. 2021. Knowledge enhanced masked language model for stance detection. In *Proceedings of NAACL*.

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*.

Taewoon Kim and Piek Vossen. 2021. Emoberta: Speaker-aware emotion recognition in conversation with roberta. *arXiv preprint arXiv:2108.12009*.

M Kowsher, Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Banglabert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, 10:91855–91870.

Jitin Krishnan, Antonios Anastasopoulos, Hemant Purohit, and Huzefa Rangwala. 2022. Cross-lingual text classification of transliterated hindi and malayalam. In *Proceedings of Big Data*.

Tarald O Kvålseth. 1989. Note on cohen's kappa. *Psychological reports*, 65(1):223–226.

Bing Liu. 2020. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge university press.

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. *CoRR*, abs/1907.11692.

Daniele Mazzocchi. 2012. langdetect: Language detection library. Python library.

Golam Sarwar Md Mursalin, Suborno Deb Bappon, and Muhammad Ibrahim Khan. 2022. A deep learning approach for recognizing textual emotion from bengali-english code-mixed data. In *Proceedings of ICCIT*.

Pieter Muysken. 2000. The study of code-mixing. *Bilingual Speech: A Typology of Code-Mixing*, 110.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of WILDRE*.

Nick Doiron. 2023. hindi-bert. Accessed: 2023-09-10.

Jianzhi Nie. 2023. Awesome instruction datasets. Accessed: 2023-09-10.

OpenAI. 2023. Gpt-3.5 turbo fine-tuning and api updates. Accessed: 2023-08-28.

Md Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023a. SentMix-3L: A novel code-mixed test dataset in bangla-english-hindi for sentiment analysis. In *Proceedings of SEALP*.

Md Nishat Raihan, Umma Tanmoy, Anika Binte Islam, Kai North, Tharindu Ranasinghe, Antonios Anastasopoulos, and Marcos Zampieri. 2023b. Offensive language identification in transliterated and code-mixed bangla. In *Proceedings of BLP*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of EMC2*.

Sebastin Santy, Anirudh Srinivasan, and Monojit Choudhury. 2021. Bertologicomix: How does code-mixing interact with multilingual bert? In *Proceedings of AdaptNLP*.

Bhaskara Hanuma Vedula, Prashant Kodali, Manish Shrivastava, and Ponnurangam Kumaraguru. 2023. Precogiiith@ wassa2023: Emotion detection for urdu-english code-mixed text. In *Proceedings of WASSA*.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in hindi-english code-mixed data: A transformer based approach. *arXiv preprint arXiv:2102.09943*.

A. Examples of Misclassified Instances

Happy: Finally got the চাবি to our new বাড়ি! So excited to start making স্মৃতি in our new space. #HomeSweetHome #NewBeginnings মैं उस अद्भुत सपोर्ट सिस्टम के लिए आभारी हूं जिसने इस यात्रा के माध्यम से मेरी मदद की। हमेशा मेरे लिए रहने के लिए धन्यवाद। #आभारी #धन्य आई एम थ्रिलड तो एनाउन्स द्याट आई ह्याव अफिसियालि कमप्लेटेड माई मेडिटेशन चाल्लेञ्ज! फिलिंग मारे सेन्टारेड एन्ड ग्राउन्डेड द्यान एभार बहरेर पर बहर saving and budgeting करार पर, आमि घाषेणा करते पेरे रामोञ्चित যে আমি আমার ছাত্র ঋণ পরিশাধে করেছি! #Debtfree #FinancialFreedom main kee apanee haal kee yaatra par kee gaee avishvasaneey yaadon ke lie bahut aabhaaree hoon. vaapas jaane ke lie intajaar nahin kar sakata! #travailgoals #advainturai

Suprise: একটি শান্তিপূর্ণ সমুদ্র সৈকতে হাঁটা, আপনার পায়ের আঙ্গুলের মধ্যে বালি অনুভব করা এবং ঢেউয়ের শব্দর মধ্যে something magical ফিল্মন অর পারিযন্ন কী কাহানিযন্ন কী সামগ্রী হয, লেকিন বাস্তবিক জীবন মেইন বহি সাকাতী হাই সোচ में खोए हुए समुद्र तट पर चलने की कल्पना करें, जब कुछ आपकी नज़र में आ जाए। Hidden treasures are waiting to be discovered, যদি আমাদের চাখে থাকে তাদের দেখার ajke somudror tire akta sundor hater kacher kaner dul peyechi

Neutral: ইমেল চেক while riding the tram to the office. ব্যস্ত ট্রেন স্টেশন পর भीड़ के माध्यम से नेविगेट करना। ওয়াচিং এ মুভি ও আ ট্যাবলেট ডুরিং টি বাস রাইড টু ওয়ার্ক। ব্যস্ত train স্টেশনে ভিড়ের মধ্য দিয়ে নেভিগেট করা। paark kee bench par baithakar bas ke aane ka intajaar kar rahe the.

Sad: একটি সম্পর্কের সমাপ্তি এবং ভালবাসা হারানারে মত situation মেনে নেয়ার মত না It makes you lonely আপকে জিবান মে হাজার লগ হনেকি বাদবি আপ ওহি এক ইনসান ক ইয়াদ কারংগি হার ওয়াক্ত নুকসান का शोक मनाना और ठीक होने के लिए आवश्यक समय लेना ठीक है। Memories flood our minds, and we find ourselves yearning for something যা আমরা আর পাব না। jake valobaschi se amar sathe sob somoy thakbe na aita kokhon o vabini

Angry: আই এম এংরি রাইট নাউ বিকজ ই জাস্ট রেসিভড আ প্যাসিভ-এগ্রেসিভ কমেন্ট ফ্রম সামওয়ান! এটা অবিশাষ্য যে কিভাবে some man how অভদ্র এবং অসম্মানজনক হতে পারে। I don't deserve to be আচরণ করার যাগ্যে way and I won't stand for it. यदि आपको मुझसे कोई समस्या है, तो मेरी पीठ पीछे भद्दी टिप्पणि करने के बजाय इसे सीधे संबोधित करने की शालीनता रखें। main kisee aur kee nakaaraatmakata ko mujhe neeche laane se mana karata hoon, lekin gambheerata se, bada hokar seekhata hoon ki ek paripakv vayask kee tarah kaise sanvaad karana hai.

Findings of the WILDRE Shared Task on Code-mixed Less-resourced Sentiment Analysis for Indo-Aryan Languages

Priya Rani[1], Gaurav Negi[2], Saroj Kumar Jha[3], Shardul Suryawanshi[2], Atul Kr. Ojha[2], Paul Buitelaar[2], John P. McCrae[2]

[1]SFI Centre for Research and Training in AI, Data Science Institute, University of Galway, Ireland
[2]Insight SFI Research Centre for Data Analytics, Data Science Institute, University of Galway, Ireland
[3]Indian Institute of Technology, Patna, India
{priya.rani, gaurav.negi, shardul.suryawanshi, atulkumar.ojha, paul.buitelaar, john.mccrae}@insight-centre.org
bhu.saroj2012@gmail.com

Abstract

This paper describes the structure and findings of the WILDRE 2024 shared task on code-mixed less-resourced sentiment analysis for Indo-Aryan Languages. The participants were asked to submit the test data's final prediction on CodaLab. A total of fourteen teams registered for the shared task. Only five participants submitted the system for evaluation on CodaLab, with only two teams submitting the system description paper. All the submitted systems exceed baseline scores, with the best F1 Scores of 0.97, 0.54, 0.45, 0.60 and 0.49 for Bangla-Hindi-English, Hindi-English, Magahi-Hindi-English, Combined, and Maithili-Hindi-English, respectively. This significant improvement from the baseline score highlights notable progress in the performance of the systems. This underscores the advancement and refinement of methodologies, highlighting the potential for further innovation for code-mixed tasks.

Keywords: Codemixing, Sentiment, Indian languages, Closely-related languages, Less-resourced languages

1. Introduction

Code-mixing, the dynamic interplay of multiple languages within a single discourse, is a widespread linguistic phenomenon observed in multilingual societies. Code-mixing is particularly intriguing when observed in closely-related languages (Rani et al., 2022). In such linguistic scenarios, where language boundaries are blurred, code-mixing becomes a dynamic expression of linguistic fluidity. Closely-related languages share lexical and syntactic similarities, allowing for seamless transitions between them in communication. This phenomenon reflects the intertwined linguistic histories and presents a rich tapestry of expression (Jain and Cardona, 2007). The nuances of code-mixing in closely-related languages highlight the intricate ways in which linguistic diversity is woven into everyday discourse, showcasing the versatility and adaptability of language in diverse linguistic landscapes. The pervasive use of the Internet and social media platforms has led to the digital availability of most languages. This digital accessibility has paved the way for a myriad of artificial intelligence (AI) applications (Goswami et al., 2020). Among these applications, sentiment analysis, machine translation, and hateful content detection stand out. Despite the increasing digital availability of languages due to the Internet and social media, the need for curated datasets for developing AI applications in many languages remains a significant challenge. Notably, numerous Indo-Aryan languages have been underrepresented in terms of linguistic resources (Winata et al., 2023). In recent years, demand has increased to create code-mixed and under-resourced Indo-Aryan languages. However, the effectiveness of existing natural language processing (NLP) techniques in utilizing these datasets and the need for novel techniques present key research areas. Understanding the applicability of current NLP methods and innovating new approaches will be crucial in maximizing the potential impact of these datasets across a spectrum of AI applications.

Sentiment analysis is a classic challenge in computational linguistics, demonstrating a profound impact on real-world applications. While sentiment analysis as a field has been expanding, and numerous shared tasks have been organised from time to time, some of them are Patra et al. (2015) organised the shared task to determine sentiment (positive, negative and neutral) of the text in three languages Bengali, Hindi and Tamil., Patwa et al. (2020) organised SemEval-2020 Task 9 on Sentiment Analysis of Code-Mixed Tweets (SentiMix 2020). The shared task provided code-mixed corpora for Hindi-English and Spanish-English annotated with word-level language identification and sentence-level sentiment labels. The shared task best teams scored 75.0% F1 score for Hinglish and 80.6% F1 for Spanglish. The shared task also reported that the BERT-like models and ensemble methods are the most common and successful approaches used by the participants. Some other shared task organised on Indian languages

WILDRE-7 Workshop, pages 17–23

are Dravidian-CodeMix shared task organised by Chakravarthia et al. (2021), shared task on sentiment analysis in Tamil and Tulu by (B et al., 2023) with the best score top system for code-mixed Tamil and Tulu texts scored macro average F1 scored by the participants are 0.32, and 0.542 respectively and so on. However, none of these shared tasks focused on code-mixed, closely-related low-resource Indo-Aryan languages. Systems have made remarkable progress in setting new performance standards, but the effectiveness of sentiment prediction in the context of code-mixed data still needs to be improved (Goswami et al., 2020). This limitation is primarily attributed to the variability in language availability and the quality of training data, which directly impacts the precision of sentiment analysis.

Overcoming the necessary gap for research in closely-related code-mixed languages, we organised this shared task on code-mixed less-resourced sentiment analysis for Indo-Aryan languages. This shared task addresses the complexities of code-mixed data from less-resourced similar languages and focuses on sentiment analysis. The task builds on code-mixed sentiment analysis but introduces language pairs and triplets of less-resourced closely related languages, Magahi-Hindi-English, Maithili-Hindi, Bangla-English-Hindi, and Hindi-English. These four languages come from the Indo-Aryan language family and are spoken in eastern India. Historically and typologically, Bangla, Maithili and Magahi belong to the same sub-branch of Indo-Aryan languages and share various lexical and linguistic features with each other (Chatterji, 1926). However, most of the time, these languages are being code-mixed with Hindi as it is the dominant language spoken in the area. Considering the challenges of processing closely related languages in code-mixed and low-resourced settings, the shared focus was letting the participants explore different machine learning and deep learning approaches to train the model on the training and validation dataset. The shared task also contributes to developing the corpora for lesser-known languages like Magahi and Maithili compared to Hindi and Bangla. This task will allow the participants to use any approach to train their model that is robust enough to perform well on a closely related code-mixed language dataset. This would also allow us to understand the language representation in various code-mixed settings and the speakers' preference of language to express their emotions in each language pair.

2. Shared-Task Setup and Schedule

This section describes the execution of the shared task. Researchers were asked to register their teams based on a detailed call for participation on our GitHub. The registered participants were able to access the dataset from our GitHub page, which included a detailed description of the format and the statistics of the dataset for each track in the task. The participants were also allowed to use additional data to train the systems, with the condition that the additional data set should be publicly available and to provide a proper citation of the data used to develop their models.

The shared task consists of two tracks described below:

1. Track 1: Given training and validation data to determine the comment's polarity, i.e., positive, negative, neutral or mixed in the same code-mixed setting. The code-mixed settings are:
    - Bangla-Hindi-English
    - Hindi-English
    - Magahi-Hindi-English
    - Combined all the language pairs
2. Track 2: Given unlabelled test data for the code-mixed Maithili language (Maithili-Hindi-English), leverage any or all of the training dataset from Track 1 to determine the sentiment of a comment in the target language.

The shared task was hosted on CodaLab[1]. Each team was allowed to submit any number of systems for evaluation, and the final ranking presented in the report includes the best-submitted system of each team. The participants were free to participate in one or both tracks and one or more of the settings of Track 1. The complete schedule of the shared task is given in Table 1.

Date	Event
22 December 2023	Registration opens
10 January 2024	Release of training data
15 February 2024	Release of test data
25 February 2024	System submission due
29 February 2024	Submission result announcement
18 March 2024	System description paper due
28 March 2024	Paper notification due

Table 1: WILDRE-7 Shared Task on Code-mixing Schedule

3. Datasets

This section presents the background information about the languages and datasets featured

[1] https://codalab.lisn.upsaclay.fr/competitions/17766

in the shared task for the two tracks. The WILDRE shared task on code-mixed less-resourced sentiment analysis for Indo-Aryan languages covers four languages, each spoken in the eastern part of India, Bangladesh and Nepal. The dataset includes a code-mixed dataset of Bangla-Hindi-English, Magahi-Hindi-English, Hindi-English and Maithili-Hindi-English. All four languages are closely related, with Bangla, Magahi, and Maithili being the least-resourced languages and Hindi being the highest-resourced language. The detailed descriptions of each of the datasets are given below:

- **Bangla-Hindi-English:** We use SentiMix-3L dataset (Raihan et al., 2023) for the first setting of Track 1. This is a trilingual code-mixed dataset between Bangla, Hindi and English for sentiment analysis. The sentiment in the dataset is classified into three categories, i.e., Positive, Negative and Neutral. Raihan et al. (2023) elaborates further details regarding the dataset's characteristics.

- **Magahi-Hindi-English:** The dataset used for the task was extracted from YouTube channels, and the data characteristics are described by (Rani et al., 2024). The dataset is annotated with four sentiment labels: positive, negative, neutral and mixed.

- **Maithili-Hindi-English:** Maithili is a less-resourced language spoken in eastern parts of India and some parts of Nepal (Jain and Cardona, 2007). Although Maithili is India and Nepal's official (scheduled) language and has about 22 million speakers, they still need more linguistic resources [2]. Therefore, we collected the data for the shared task from YouTube's different channels. These channels' contents consist of various genres like entertainment, Politics, Environment, debates, general histories, general knowledge and many more. Later on we annotated the data for sentiment analysis using the same annotation guidelines as Magahi data (Rani et al., 2024) with the inter-annotator agreement of 0.73 using Cohen's Kappa [3].

- **Hindi-English**. Similar to Magahi and Maithili data, Hindi-English data was also collected from YouTube Channels and was annotated along with Magahi and Maithili Data annotation.

[2] https://en.wikipedia.org/wiki/Maithili_language

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html

The complete shared task datasets are available at GitHub.[4]. The detailed statistics of the dataset in each language are provided in Table 2.

Language sets	Training	Validation	Test
	Trac 1		
Bangla-Hindi-English	703	151	151
Magahi-Hindi-English	865	185	185
Hindi-English	2507	537	537
Combined	4075	873	873
	Trac 2		
Maithili-Hindi-English	–	–	263

Table 2: Detailed statistics of the dataset

4. Method

4.1. Evaluation

In assessing the efficacy of the multi-class classification approach, we employ the macro-average F1-score. This metric is particularly advantageous in scenarios where sentiment class distributions are imbalanced, as it accords equal weight to each sentiment class's contribution. By computing the F1-score for each sentiment class independently and then averaging these scores, the macro-average F1-score offers a comprehensive and unbiased reflection of the model's performance across all sentiments. Consequently, this measure ensures that the model's efficiency is not disproportionately influenced by the more prevalent sentiments in the datasets, thereby providing a holistic view of its classification capabilities. The evaluation was performed in two different Tracks:

Track 1: The macro-averaged F1-score is calculated on the test split of the dataset for the following language mixes for which the training and validation datasets were made available:

- Hindi-English
- Magahi-Hindi-English
- Bangla-English
- Combined all the language pairs (1+2+3)

Track 2: The macro-averaged F1-score is calculated for code-mixed Maithili language (Maithili-Hindi-English). This was a zero-shot evaluation, as the training data was not provided.

[4] https://github.com/wildre-workshop/wildre-7_code-mixed-sentiment-analysis

4.2. Baseline

We started with a simple baseline. The baseline model has an embedding layer. Each token/word is mapped to a vector of length 300. It is followed by an LSTM (Bi-LSTM) layer having 64 recurrent units. It is followed by two dense layers of 128 and 3 units, respectively. For the baselines, we do not use pre-trained word embeddings. The embedding layer is trained with the classification model.

5. Submitted Systems

A total of 14 teams registered for the shared task. Out of the 14 registered teams, five teams successfully submitted their systems. Most teams submitted the systems for each language set in both tracks except one team that participated only in track 1, Hindi-English and Magahi-Hindi-English language sets. Finally, all the submitted systems comprehensively utilized LLMs due to their versatility in the NLP tasks (Brown et al., 2020). The use of open-source LLMs like Mistral(Jiang et al., 2023, 2024), Llama(Touvron et al., 2023) and Gemma(Team et al., 2024) showcases the capability of open-source freely available LLMs for less-resourced language research.

Teams	BHE	HE	MHE	Combined	MaHE	System Description
FZZG	Yes	Yes	Yes	Yes	Yes	Yes
pakkapro	Yes	Yes	Yes	Yes	Yes	No
kriti7	No	Yes	Yes	No	No	No
hkesevam	Yes	Yes	Yes	Yes	Yes	No
MLInitiative	Yes	Yes	Yes	Yes	Yes	Yes
Total	4	5	5	4	4	2

Table 3: Details of the participated teams in the WILDRE 2024 Shared Task

5.1. Team FZZG

The system used by the `Team FZZG` was the best performing in all the sub-tasks in both tracks (Thakkar et al., 2024). The used Mixtral-8x7B model (Jiang et al., 2024). They used LoRA (Hu et al., 2022) to fine-tune the 4-bit quantized language model in a parameter-efficient manner. The fine-tuning process used a predefined instruction format from the Alpaca dataset (Taori et al., 2023).

Instruction: Classify the given article as either positive or negative or neutral or mix sentiment.
alpaca_prompt: """
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
Instruction:
{}
Input:
{}
Response:
{}
"""

Prompt 1: Instruction for Mixtral-8x7B Model

They also performed preliminary experiments with XLM-RoBERTa (Conneau et al., 2020) in addition to the Mixtral-8x7B model which they ended up selecting. In addition to the training dataset released in the shared task, they utilized SentMix-3L Bangla-English-Hindi code-mixed dataset (Raihan et al., 2023).

5.2. Team MLInitiative

The MLInitiative system was designed based on a multi-step approach for code-mixed sentiment prediction (Veeramani et al., 2024). The first step in this multi-step system is used to generate additional input features for the LLM that makes the final prediction. The additional features include:

- Decomposed Language Inputs: The code-mixed input is decomposed and separated into individual languages. They are extracted with three LLMs, i.e. Mistral(Jiang et al., 2023), Llama(Touvron et al., 2023) and Gemma(Team et al., 2024).
- Named Entity Extraction: Named entities are extracted from the code-mixed texts with mBERT (Devlin et al., 2019) model.
- Preliminary Label Prediction: mBERT is used to predict the sentiments on the code-mixed text inputs.

In the final step, all the features are fed to the LLM to obtain the final predictions. They experimented with three different language models and found variable efficiency of models in different code-mixed settings.

6. Results

Participants were instructed to submit their output files for our CodaLab competition in ZIP format. Each submission was packaged in a ZIP file, which included a CSV file containing the text_id

and the corresponding generated sentiment labels, along with a text file detailing the trained models used. The files were required to be named following the format: **team_name_language**. For each language track, participants submitted a single ZIP file structured as described above. The results of all the participating teams are summarized in Table 4.

Team	Task	F1-Score	Precision	Recall
Track 1				
BASELINE	Bangla-English	0.34	0.34	0.34
FZZG(Mixtral)	Bangla-English	0.97	0.97	0.97
MLInitiative(Mistral)	Bangla-English	0.67	0.76	0.68
BASELINE	Hindi-English	0.24	0.24	0.24
FZZG(Mixtral)	Hindi-English	0.54	0.54	0.56
MLInitiative(Gemma)	Hindi-English	0.34	0.35	0.39
BASELINE	Maghi-Hindi-English	0.21	0.18	0.25
FZZG(Mixtral)	Maghi-Hindi-English	0.45	0.44	0.57
MLInitiative(Gemma)	Maghi-Hindi-English	0.26	0.28	0.27
BASELINE	Combined	0.29	0.28	0.29
FZZG(Mixtral)	Combined	0.60	0.64	0.57
MLInitiative(Gemma)	Combined	0.35	0.36	0.36
Track 2				
BASELINE	Maithili-Hindi-English	0.17	0.24	0.22
FZZG(Mixtral)	Maithili-Hindi-English	0.49	00.45	0.59
MLInitiative(Llama)	Maithili-Hindi-English	0.35	0.36	0.36

Table 4: System Evaluation

7. Discussion

After analysing the shared task results, we made a few interesting observations. First, data scarcity does impact training on classification tasks, as we can see the difference in results of the two teams mentioned in table 4, where `Team FZZG` trained their model on extra data other than the data provided in the shared task whereas, `Team MLInitiative` trained only on the data provided in the shared task. However, balanced data could mitigate potential issues, as demonstrated by the outcomes of the Bangla-Hindi-English task in contrast to another language. Distribution of the data for Bangla-Hindi-English (Raihan et al., 2023) is pretty balanced compared to other languages (Rani et al., 2024).

The findings demonstrate that Large Language Models (LLMs) significantly surpass a basic benchmark in predicting sentiment in code-mixed text. This indicates that LLMs possess a robust capability to analyze and interpret the sentiment of text that blends multiple languages, which is a complex challenge in computational linguistics.

`Team MLInitiative` augmented their model's input by incorporating decomposed linguistic elements, extracting named entities, and integrating secondary classification outcomes. These refinements and a sophisticated model architecture contributed significantly to the model's performance, surpassing baseline metrics.

`Team FZZG` integrated all the code-mixed training datasets into a single training dataset. Subsequently, the fine-tuned model using this integrated dataset demonstrated superior performance compared to models trained on the individual code-mixed datasets. This outcome suggests that models can learn transferable features from closely-related code-mixed language pairs, enhancing their ability to analyze sentiments.

8. Conclusion

In this paper, we report the findings of the WILDRE-7 shared task on code-mixed less-resourced sentiment analysis for Indo-Aryan languages. All the systems submitted used large language models to solve the problems of sentiment analysis in closely related code-mixed scenarios in low-resource settings. The baselines were trained on Bi-LSTM models to allow the participants to explore and experiment with any deep-learning techniques to find the best solution to the task. The `Team FZZG` scored the best in all the tasks. Nonetheless, the collective efforts of both teams contribute towards the understanding of approaches that enhance the efficacy of sentiment analysis systems in the less-resourced code-mixed setting.

9. Acknowledgement

This shared task was in part supported by Science Foundation Ireland under grant number SFI/18/CRT/6223 (Centre for Research Training in Artificial Intelligence) and co-funded by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2 Insight_2, Insight SFI Research Centre for Data Analytics.

10. References

Premjith B, Jyothish Lal G, Sowmya V, Bharathi Raja Chakravarthi, Rajeswari Natarajan, Nandhini K, Abirami Murugappan, Bharathi B, Kaushik M, Prasanth Sn, Aswin Raj R, and Vijai Simmon S. 2023. Findings of the shared task on multimodal abusive language detection and sentiment analysis in Tamil and Malayalam. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 72–79, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher

Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bharathi Raja Chakravarthia, Ruba Priyadharshinib, Sajeetha Thavareesanc, Dhivya Chinnappad, Durairaj Thenmozhie, Elizabeth Sherlyf, John P McCraea, Adeep Handeh, Rahul Ponnusamy, Shubhanker Banerjeej, and Charangan Vasantharajan. 2021. Findings of the sentiment analysis of Dravidian languages in code-mixed text. In *Proceedings of the 13th Forum for Information Retrieval Evaluation*.

Suniti Kumar Chatterji. 1926. *Origin and development of the Bengali language*. Calcutta University Press.

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, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. 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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Koustava Goswami, Priya Rani, Bharathi Raja Chakravarthi, Theodorus Fransen, and John P. McCrae. 2020. ULD@NUIG at SemEval-2020 task 9: Generative morphemes with an attention model for sentiment analysis in code-mixed text. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 968–974, Barcelona (online). International Committee for Computational Linguistics.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Danesh Jain and George Cardona. 2007. *The Indo-Aryan Languages*. Routledge.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. *Shared Task on Sentiment Analysis in Indian Languages (SAIL) Tweets - An Overview*, page 650–655. Springer International Publishing.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790, Barcelona (online). International Committee for Computational Linguistics.

Md Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. Sentmix-3l: A novel code-mixed test dataset in bangla-english-hindi for sentiment analysis. In *Proceedings of the First Workshop in South East Asian Language Processing*, pages 79–84.

Priya Rani, John P. McCrae, and Theodorus Fransen. 2022. MHE: Code-mixed corpora for similar language identification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3425–3433, Marseille, France. European Language Resources Association.

Priya Rani, Gaurav Negi, Theodorus Fransen, and John P. McCrae. 2024. Macms: Magahi code-mixed dataset for sentiment analysis. *arXiv preprint arXiv:2403.04639*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Gaurish Thakkar, Marko Tadić, and Nives Mikelic Preradovic. 2024. Fzzg at wildre-7: Fine-tuning pre-trained models for code-mixed, less-resourced sentiment analysis. In *Proceedings of the 7th Workshop on Indian Language Data: Resources and Evaluation @LREC-COLING-2024 (WILDRE-7)*, Turin, Italy. ELRA Language Resources Association and the International Committee on Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2024. Mlinitiative@wildre7: Hybrid approaches with large language models for enhanced sentiment analysis in code-switched and code-mixed texts. In *Proceedings of the 7th Workshop on Indian Language Data: Resources and Evaluation @LREC-COLING-2024 (WILDRE-7)*, Turin, Italy. ELRA Language Resources Association and the International Committee on Computational Linguistics.

Genta Winata, Sudipta Kar, Marina Zhukova, Thamar Solorio, Mona Diab, Sunayana Sitaram, Monojit Choudhury, and Kalika Bali, editors. 2023. *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Singapore.

Multilingual Bias Detection and Mitigation for Indian Languages

Ankita Maity*, **Anubhav Sharma***, **Rudra Dhar***, **Tushar Abhishek**† *
Manish Gupta† *, **Vasudeva Varma***
*IIIT Hyderabad, India
†Microsoft, India

Abstract

Lack of diverse perspectives causes neutrality bias in Wikipedia content leading to millions of worldwide readers getting exposed by potentially inaccurate information. Hence, neutrality bias detection and mitigation is a critical problem. Although previous studies have proposed effective solutions for English, no work exists for Indian languages. First, we contribute two large datasets, mWikiBias and mWNC, covering 8 languages, for the bias detection and mitigation tasks respectively. Next, we investigate the effectiveness of popular multilingual Transformer-based models for the two tasks by modeling detection as a binary classification problem and mitigation as a style transfer problem. We make the code and data publicly available.

Keywords: Neutral Point of View, Bias Detection, Bias Mitigation, Indian language NLG, Transformer Models

1. Introduction

Wikipedia has three core content policies: Neutral Point of View (NPOV), No Original Research, and Verifiability[1]. NPOV means that content should represent fairly, proportionately, and, as far as possible, without editorial bias, all the significant views that have been published by reliable sources on a topic[2]. This means (1) Opinions should not be stated as facts and vice versa. (2) Seriously contested assertions should not be stated as facts. (3) Nonjudgmental language should be preferred. (4) Relative prominence of opposing views should be indicated. This is definition of bias we follow in this paper.

Considering Wikipedia's (1) volume and diversity of content, (2) frequent updates, and (3) large and diverse userbase, automatic bias detection and suggestion of neutral alternatives is important. Bias can lead to inaccurate information or dilution of information. Particularly, lower article quality and fewer editors of Indian language Wikipedia pages makes such a system indispensable. Hence, in this work, we study how to detect sentences that violate the NPOV guidelines and convert them to more neutral sentences for Indian languages, as shown in Fig. 1.

While there exists bias detection and mitigation studies (Zhong et al., 2021; Pryzant et al., 2020; Lai et al., 2022; Liu et al., 2021) for English, there is hardly any such work for other languages. Aleksandrova et al. (Aleksandrova et al., 2019) work on bias detection for Bulgarian and French, but their method requires a collection of language-specific NPOV tags; and is therefore difficult to extend to Indian languages. Lastly, there exists no dataset for multilingual bias mitigation. We fill the gap in this paper by proposing two new multilingual bias detection and mitigation datasets, mWikiBias and mWNC, each covering 8 languages: English (en) and seven Indian languages - Hindi (hi), Marathi (mr), Bengali (bn), Gujarati (gu), Tamil (ta), Telugu (te) and Kannada (kn).

Bias detection is challenging because certain words lead to bias if they are written in some contexts, while not in other contexts. For bias detection, we perform binary classification using MuRIL (Khanuja et al., 2021), InfoXLM (Chi et al., 2021) and mDeBERTa (He et al., 2022) in zero-shot, monolingual and multilingual settings. Bias mitigation is challenging because of subjectivity and context-dependence, and the models need to strike a good balance between fairness and content preservation. For bias mitigation, we perform style transfer using IndicBART (Dabre et al., 2022), mT0 (Muennighoff et al., 2023) and mT5 (Xue et al., 2021). These models provide strong baseline results for the novel multilingual tasks.

Overall, we make the following contributions in this paper.

- We propose multilingual bias detection and mitigation for Indian languages.
- We contribute two novel datasets, mWikiBias and mWNC, to multilingual natural language generation (NLG) community. Across 8 languages, they contain ∼568K and ∼78K samples for bias detection and mitigation resp.
- Extensive experiments show that mDeBERTa outperforms MuRIL and InfoXLM for the bias detection task. On the other hand, mT5 and

[1] https://en.wikipedia.org/wiki/Wikipedia:Core_content_policies
[2] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

WILDRE-7 Workshop, pages 24–29

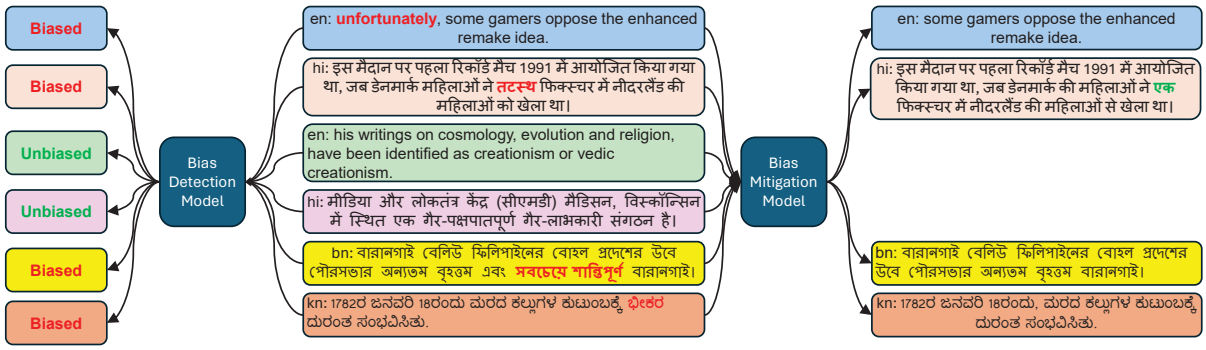


Figure 1: Bias Detection and Mitigation Examples from mWikiBias Dataset

mT0 perform the best for bias mitigation on mWikiBias and mWNC respectively.

2. Related Work

Several kinds of societal biases have been studied in the literature as part of responsible AI model building (Sheng et al., 2021; Badjatiya et al., 2019): promotional tone (De Kock and Vlachos, 2022), puffery (Bertsch and Bethard, 2021), political bias (Fan et al., 2019), and gender and racial bias (Field et al., 2022; Parikh et al., 2021). In this work, we focus on a more general form of bias called as neutrality bias. Earlier work on neutrality bias detection leveraged basic linguistic features (Recasens et al., 2013; Hube and Fetahu, 2018) while recent work uses Transformer based models (Pryzant et al., 2020; Zhong et al., 2021). Unfortunately, these studies (Recasens et al., 2013; Pryzant et al., 2020; Zhong et al., 2021; Hube and Fetahu, 2018) focus on English only. Aleksandrova et al. (Aleksandrova et al., 2019) work on bias detection for Bulgarian and French, but their method requires a collection of language-specific NPOV tags, making it difficult to extend to Indian languages.

Bias mitigation is under-studied even for English (Liu et al., 2021). We contribute datasets and strong initial baseline methods towards multilingual bias mitigation.

3. mWikiBias and mWNC Datasets

Popular bias detection and mitigation corpora in English like Wiki Neutrality Corpus (WNC) (Pryzant et al., 2020) and WikiBias (Zhong et al., 2021) were created by looking for NPOV-related tags in the edit history of the English Wikipedia dumps. Both datasets have parallel sentence structures (biased sentence linked with an unbiased version). Replication of their data curation pipeline for Indian languages did not work due to a lack of frequency and consistency in tag usage for edits in the revision history of corresponding Wikipedia pages.

Hence, we translated these datasets using IndicTrans (Ramesh et al., 2022) to create mWNC and mWikiBias datasets for eight languages. To create cleaner datasets, we used the following heuristics to filter samples. (1) A biased and its corresponding unbiased sentence in English typically differ by very few words. Hence, we removed samples where translation of biased sentence and unbiased sentence were exactly the same for at least one of our target languages. (2) To reduce impact of translation errors, we removed samples where sentences contained regex matches for URLs, phone numbers, and email IDs.

For every parallel translated pair of (biased, unbiased) sentence in each language l, we create one sample for bias mitigation dataset, and two samples (biased and unbiased) for bias detection dataset. Overall, the total number of samples for classification are 287.6K and 280.0K for mWikiBias and mWNC respectively. To reduce training compute, we took a random sample from the overall bias mitigation data, leading to 39.4K and 39.0K paired samples in the mWikiBias and mWNC respectively[3]. The number of samples for each language in both the datasets is consistent. Further, both of our bias detection datasets contain an equal number of biased and unbiased samples. We divide the datasets into a train/val/test split of 90/5/5.

4. Multilingual Bias Detection and Mitigation

We train multilingual bias detection and mitigation models using train part of mWikiBias and mWNC respectively. As shown in Fig. 1, these models detect whether the sentence is biased and con-

[3]Experiments with full bias mitigation dataset showed similar results.

		MuRIL				InfoXLM				mDeBERTa			
		Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
mWiki Bias	ZeroShot	59.26	61.53	59.26	57.19	59.28	59.74	59.28	58.81	60.99	61.63	60.99	60.45
	MonoLingual	62.66	65.15	62.66	60.97	60.97	62.06	60.97	60.01	64.82	65.63	64.82	64.31
	MultiLingual	65.11	**66.33**	65.11	64.41	63.42	64.55	63.42	62.64	**65.14**	65.64	**65.14**	**64.83**
mWNC	ZeroShot	63.04	64.00	63.04	62.38	62.08	62.81	62.08	61.53	63.02	64.02	63.02	62.34
	MonoLingual	64.82	65.95	64.82	64.17	63.15	63.71	63.15	62.75	66.59	66.92	66.59	66.42
	MultiLingual	66.72	**67.24**	66.72	66.46	65.49	65.75	65.49	65.34	**66.96**	67.03	**66.96**	**66.92**

Table 1: Bias Detection Results.

		MuRIL				InfoXLM				mDeBERTa			
		Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
mWikiBias	bn	64.55	**65.65**	64.55	63.92	62.17	63.38	62.17	61.30	**64.62**	65.15	**64.62**	**64.31**
	en	73.69	**74.74**	73.69	73.40	72.89	73.74	72.89	72.65	**74.19**	74.57	**74.19**	**74.09**
	gu	63.77	**64.93**	63.77	63.06	62.03	63.25	62.03	61.13	**63.91**	64.35	**63.91**	**63.63**
	hi	**65.31**	**66.37**	**65.31**	64.73	63.54	64.60	63.54	62.86	65.01	65.34	65.01	**64.82**
	kn	**64.33**	**65.63**	**64.33**	63.57	62.48	63.50	62.48	61.76	63.96	64.49	63.96	**63.64**
	mr	**62.74**	**63.98**	**62.74**	61.89	61.47	62.52	61.47	60.65	62.26	62.71	62.26	**61.92**
	ta	63.05	64.47	63.05	62.12	61.99	63.23	61.99	61.07	**63.80**	**64.55**	**63.80**	**63.33**
	te	**63.46**	**64.90**	**63.46**	62.56	60.81	62.17	60.81	59.69	63.34	63.99	63.34	**62.91**
	avg	65.11	**66.33**	65.11	64.41	63.42	64.55	63.42	62.64	**65.14**	65.64	**65.14**	**64.83**
mWNC	bn	**66.75**	**67.34**	**66.75**	**66.47**	65.01	65.32	65.01	64.83	66.46	66.53	66.46	66.42
	en	71.08	71.43	71.08	70.96	71.57	71.66	71.57	71.54	**72.92**	**72.92**	**72.92**	**72.91**
	gu	66.00	**66.48**	66.00	65.76	64.33	64.63	64.33	64.15	**66.42**	66.46	**66.42**	**66.40**
	hi	67.13	**67.53**	67.13	66.95	66.28	66.44	66.28	66.20	**67.45**	67.47	**67.45**	**67.44**
	kn	66.40	**66.92**	66.40	66.14	64.77	65.04	64.77	64.61	**66.55**	66.61	**66.55**	**66.52**
	mr	**64.90**	**65.48**	**64.90**	**64.57**	63.70	63.94	63.70	63.54	64.44	64.56	64.44	64.37
	ta	**65.70**	**66.29**	**65.70**	65.38	64.36	64.69	64.36	64.15	65.68	65.83	65.68	**65.60**
	te	**65.78**	**66.43**	**65.78**	65.44	63.93	64.30	63.93	63.69	65.75	65.87	65.75	**65.68**
	avg	66.72	**67.24**	66.72	66.46	65.49	65.75	65.49	65.34	**66.96**	67.03	**66.96**	**66.92**

Table 2: Detailed Language-wise Bias Detection Results for Multilingual Setup.

vert it to a more neutral sentence if bias is detected. So, for example, the Hindi sentence मीडिया और लोकतंत्र केंद्र (सीएमडी) मैडिसन, विस्कॉन्सिन में स्थित एक गैर-पक्षपातपूर्ण गैर-लाभकारी संगठन है (the Center for Media and Democracy (CMD) is a non-partisan non-profit organization based in Madison, Wisconsin) is detected as an unbiased sentence, while the Bengali sentence বারানগাই বেলিউ ফিলিপাইনের বোহল প্রদেশের উবে পৌরসভার অন্যতম বৃহত্তম এবং সবচেয়ে শান্তিপূর্ণ বারানগাই (Barangay Benliw is one of the largest and the most peaceful Barangay in the municipality of Ubay, in the province of Bohol, Philippines) is detected as biased and thus converted to a more neutral বারানগাই বেলিউ ফিলিপাইনের বোহল প্রদেশের উবে পৌরসভার অন্যতম বৃহত্তম বারানগাই (Barangay Benliw is one of the largest Barangays in the municipality of Ubay, in the province of Bohol, Philippines). Similarly, the Kannada sentence 1782ರ ಜನವರಿ 18ರಂದು ಮರದ ಕಲ್ಲುಗಳ ಕುಟುಂಬಕ್ಕೆ ಭೀಕರ ದುರಂತ ಸಂಭವಿಸಿತು (on 18 January 1782, a horrendous tragedy struck the Woodmason family) is converted to a more neutral 1782ರ ಜನವರಿ 18ರಂದು, ಮರದ ಕಲ್ಲುಗಳ ಕುಟುಂಬಕ್ಕೆ ದುರಂತ ಸಂಭವಿಸಿತು (on 18 January 1782, tragedy struck the Woodmason family).

Multilingual Bias Detection Method: For bias detection, we finetune three Transformer encoder-only multilingual models: InfoXLM (Chi et al., 2021), MuRIL (Khanuja et al., 2021), and mDeBERTa (He et al., 2022), with a twin linear layer setup to detect whether a sentence is biased. We experiment with three different training setups: (1) zero-shot (training only on English and testing on the other languages), (2) monolingual (one language at a time) and (3) multilingual (trained on all languages together). For fair comparisons, we use 12 layer models with a dimensionality of 768.

Multilingual Bias Mitigation Method: For bias mitigation, we finetune three multilingual encoder-decoder transformer-based models: mT5 (Xue et al., 2021), IndicBART (Dabre et al., 2022), and mT0 (Muennighoff et al., 2023) over the parallel corpora to perform debiasing. For fair comparisons, we use the small version of all three models for our experiments.

Metrics: We evaluate bias detection models using four popular binary classification metrics: accuracy (Acc), macro-precision (P), macro-recall (R) and macro-F1.

Effectiveness of bias mitigation models should be evaluated broadly on two aspects: match with groundtruth and debiasing accuracy. For measuring match with groundtruth unbiased sentences, we use standard NLG metrics like BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), chrF (Popović, 2015) and BERTScore (Zhang et al., 2019). We measure debiasing accuracy using "Normalized Accuracy (NAcc)" defined as follows. Let N be the percent of ground truth sentences in the test set that are classified as "unbiased" by our best bias detection model. First, given a bias mitigation model, we compute the percent of its generated outputs that are classified as "unbiased" by our best bias detection model. Sec-

		MonoLingual						MultiLingual					
		B	M	C	BS	NAcc	HM	B	M	C	BS	NAcc	HM
mWiki Bias	IndicBART	**63.67**	75.87	80.04	91.58	71.57	80.35	46.32	64.62	68.94	88.47	73.84	80.50
	mT0	61.57	77.05	80.84	93.24	76.77	84.21	60.86	77.04	80.89	93.20	77.73	84.76
	mT5	58.81	76.74	80.23	92.97	73.23	81.93	63.26	**77.41**	**81.39**	**93.40**	**79.38**	**85.82**
mWNC	IndicBART	54.98	69.25	75.27	90.99	65.52	76.18	17.58	59.67	61.15	85.54	**71.12**	77.67
	mT0	53.09	70.01	75.75	91.27	69.15	78.68	55.23	**70.61**	**76.54**	**91.50**	70.59	**79.70**
	mT5	**55.39**	70.28	76.22	91.36	66.57	77.02	55.27	70.46	76.41	91.47	69.83	79.20

Table 3: Bias Mitigation Results. B=BLEU, M=METEOR, C=chrF, BS=BERTScore, NAcc=NormAcc, HM=Harmonic Mean of BS and NAcc.

	mT5 mWikiBias						mT0 mWNC					
	B	M	C	BS	NAcc	HM	B	M	C	BS	NAcc	HM
bn	60.60	75.82	80.03	92.81	76.50	83.87	54.76	68.48	75.10	90.72	70.29	79.21
en	86.02	92.68	91.63	98.30	88.06	92.90	79.06	87.56	87.38	97.42	79.97	87.83
gu	61.35	76.39	79.54	92.85	78.02	84.79	55.47	69.56	74.73	90.79	67.34	77.32
hi	69.36	82.87	82.76	94.16	75.78	83.97	63.12	76.81	77.85	92.19	69.49	79.25
kn	60.84	75.63	82.05	93.28	78.31	85.14	54.69	68.88	77.55	91.40	67.26	77.49
mr	58.19	73.25	78.11	92.12	79.43	85.31	50.24	65.44	72.65	89.77	68.78	77.89
ta	53.03	69.70	78.15	91.57	80.26	85.54	35.66	61.91	72.51	89.37	71.66	79.54
te	56.72	72.97	78.86	92.11	78.67	84.86	48.84	66.21	74.55	90.31	70.00	78.87
avg	63.26	77.41	81.39	93.40	79.38	85.82	55.23	70.61	76.54	91.50	70.59	79.70

Table 4: Detailed Language-wise Bias Mitigation Resultsfor the best models per dataset. B=BLEU, M=METEOR, C=chrF, BS=BERTScore, NAcc=NormAcc, HM=Harmonic Mean of BS and NAcc.

ond, we normalize this quantity by N and call the ratio as Normalized Accuracy (NAcc).

A model can easily obtain high match with groundtruth by simply copying words from the input. Similarly, a model can easily obtain high NAcc score by predicting a constant highly unbiased sentence independent of the input. A good model should be able to strike a favourable tradeoff between the two aspects. Among the four metrics for computing the match, BERT-Score has been shown to be the most reliable in NLG literature, because it captures semantic match rather than just a syntactic match. Hence, we compute the harmonic mean of BERT-Score and NAcc Score and report it as HM.

Implementation Details: For MuRIL and InfoXLM, we use a learning rate of 1e-6, weight decay of 0.001, and dropout of 0.1. We trained for 15 epochs using a batch size of 320 and mixed precision training. For mDeBERTa, we use a learning rate 2e-5 with a weight decay of 0.01, keeping the other parameters the same. We use a batch size of 12 for the bias mitigation experiments and train for 10 epochs, using early stopping with a patience of 3. We use Adafactor optimizer with a learning rate of 1e-3 for mT5 and mT0 and AdamW optimiser with a learning rate of 1e-4 for IndicBART. All models use a weight decay of 0.01. All models were trained on a machine with 4 NVIDIA V100 GPUs having 32GB of RAM.

5. Results

Bias Detection Results: We show a summary of bias detection results, averaged across the 8 languages, in Table 1 and language-wise details in Table 2. Table 1 shows that (1) Multilingual models outperform monolingual models, which in turn outperform zero-shot approaches. (2) Across both the datasets, mDeBERTa and MuRIL, both trained in a multilingual setting, exhibit the strongest performance, with mDeBERTa slightly outperforming MuRIL.

From the language-wise results in Table 2, we observe the following: (1) As expected, for both datasets, across all models and metrics, best results are for English. We also observe that the models perform the worst for Marathi and Telugu. (2) In general, MuRIL is better in terms of precision, but mDeBERTa is better in terms of recall and also F1.

Bias Mitigation Results: We show a summary of bias mitigation results in Table 3 and language-wise bias mitigation results in Table 4 using N=73.52 and 76.17 for mWikiBias and mWNC respectively. From the results, we observe that (1) Broadly, multilingual models outperform monolingual counterparts. (2) mT5 is better for mWikiBias providing a high HM of 85.82, while mT0 is better for mWNC providing a high HM of 79.70. (3) As expected, both the models work best for English.

Human Evaluation Results:

We asked 4 Computer Science bachelors students with language expertise to evaluate the generated outputs (mT5 multilingual for mWIKIBIAS and mT0 multilingual for mWNC) on 3 criteria, each on a scale of 1 to 5: fluency, whether the bias is reduced and whether the meaning is preserved when compared to input. This was done for 50 samples per language for both datasets. Table 5 shows that automated evaluation correlates well with human judgment, with English predictions

Lang.	mWikiBias			mWNC		
	Fluency (↑)	Bias (↓)	Meaning (↑)	Fluency (↑)	Bias (↓)	Meaning (↑)
bn	4.42	3.12	4.79	3.94	2.68	4.80
en	**4.92**	2.72	**4.84**	**4.86**	**2.40**	**4.92**
hi	4.20	3.20	4.76	4.60	2.64	**4.92**
te	4.40	**2.50**	4.81	3.88	2.45	4.75

Table 5: Human Evaluation Results

showing the best results. mWNC is easier for the models to debias than mWIKIBIAS. The model outputs were generally fluent and had similar content as the input text. However, a wider variance in bias mitigation abilities was observed for the 3 Indian languages tested compared to English. Ambiguity in bias assessment and noise in the reference text made ∼20% of the samples challenging for human annotators.

6. Conclusion

In this paper, we proposed the critical problems of multilingual bias detection and mitigation for Indian languages. We also contributed two data sets, mWikiBias and mWNC. We presented baseline results using standard Transformer based models. We make our code and data set publicly available[4]. In the future, we would like to experiment with reinforcement learning based methods which could use detection based scores to enhance generation.

7. Bibliographical References

References

Desislava Aleksandrova, François Lareau, and Pierre André Ménard. 2019. Multilingual sentence-level bias detection in wikipedia. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 42–51.

Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *The world wide web conference*, pages 49–59.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Amanda Bertsch and Steven Bethard. 2021. Detection of puffery on the english wikipedia. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 329–333.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, He-Yan Huang, and Ming Zhou. 2021. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh M Khapra, and Pratyush Kumar. 2022. Indicbart: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1849–1863.

Christine De Kock and Andreas Vlachos. 2022. Leveraging wikipedia article evolution for promotional tone detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5601–5613.

Lisa Fan, Marshall White, Eva Sharma, Ruisi Su, Prafulla Kumar Choubey, Ruihong Huang, and Lu Wang. 2019. In plain sight: Media bias through the lens of factual reporting. 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 6343–6349.

Anjalie Field, Chan Young Park, Kevin Z Lin, and Yulia Tsvetkov. 2022. Controlled analyses of social biases in wikipedia bios. In *Proceedings of the ACM Web Conference 2022*, pages 2624–2635.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Christoph Hube and Besnik Fetahu. 2018. Detecting biased statements in wikipedia. In *Compan-*

[4] https://github.com/Ankita-Maity/Bias-Detection-Mitigation/

ion proceedings of the the web conference 2018, pages 1779–1786.

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*.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2022. Multilingual pre-training with language and task adaptation for multilingual text style transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 262–271.

Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, Lili Wang, and Soroush Vosoughi. 2021. Mitigating political bias in language models through reinforced calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14857–14866.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pulkit Parikh, Harika Abburi, Niyati Chhaya, Manish Gupta, and Vasudeva Varma. 2021. Categorizing sexism and misogyny through neural approaches. *ACM Transactions on the Web (TWEB)*, 15(4):1–31.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 480–489.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, AK Raghavan, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, J Mahalakshmi, Divyanshu Kakwani, 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.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2021. Societal biases in language generation: Progress and challenges. *arXiv preprint arXiv:2105.04054*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yang Zhong, Jingfeng Yang, Wei Xu, and Diyi Yang. 2021. Wikibias: Detecting multi-span subjective biases in language. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1799–1814.

Dharmaśāstra Informatics: Concept Mining System for Socio-Cultural Facet in Ancient India

Arooshi Nigam, Subhash Chandra
Department of Sanskrit
University of Delhi
New Delhi-110007, India
Email: {anigam, schandra}@sanskrit.du.ac.in

Abstract

The heritage of Dharmaśāstra (DS) represents an extensive cultural legacy, spanning diverse fields such as family law, social ethics, culture and economics. In this paper, a new term "Dharmaśāstric Informatics," is proposed which leverages computational methods for concept mining to unravel the socio-cultural complexities of ancient India as reflected in the DS. Despite its profound significance, the digitization and online information retrieval of DS texts encounter notable challenges. Therefore, the primary aim of this paper is to synergize digital accessibility and information mining techniques to enhance access to DS knowledge traditions. Through the utilization of heritage computing methodologies, it is an endeavour to develop a robust system for digitizing DS texts comprehensively, facilitating instant referencing and efficient retrieval, catering to the needs of researchers and scholars across disciplines worldwide. By leveraging advanced digital technologies and the burgeoning IT landscape, it seeks to create a seamless and user-friendly platform for accessing and exploring DS texts. This experiment not only promotes scholarly engagement but also serves as an invaluable resource for individuals interested in delving into the intricate realms of archaic Indian knowledge traditions. Ultimately, our efforts aim to amplify the visibility and accessibility of DS knowledge, fostering a deeper understanding and appreciation of this profound cultural heritage.

Keywords: Information Extraction (IE), Online Indexing, Concept Mining, Heritage Computing (HC), Cultural Tradition, Dharmaśāstra (DS)

1. Background and Introduction

The DS, revered as repositories of antiquated Indian wisdom, provide invaluable glimpses into the socio-cultural landscape of ancient India. Penned in Sanskrit, these texts cover a vast array of subjects, spanning from legal doctrines, philosophical traditions and ethical precepts to societal conventions and religious ceremonies (Biswas and Banerjee, 2016). These texts constitute a fundamental aspect of classical Indian literature and are dedicated to delineating the principles and guidelines for social management and individual conduct (Phillips, 2014). As noted by Banerjee (1999), DS are uniquely focused on prescribing duties for every individual within society, outlining the ethical and moral framework for right conduct. Furthermore, Dubey (2012) underscores their significance by emphasizing their portrayal of dharma as the righteous path of living. In essence, these texts serve as comprehensive guides to ethical behaviour, cultural aspects and moral obligations, offering invaluable insights into the ancient Indian tradition of social governance and individual responsibilities. Untangling the complexities of the socio-cultural milieu encapsulated within the DS is a challenging yet profoundly enriching strive, carrying immense scholarly import.

Traditional approaches to textual analysis and interpretation have long been the cornerstone of studying the DS. Scholars meticulously pore over these texts, dissecting their verses, and deciphering their meanings to glean insights into primitive Indian society. However, the emergence of the field of informatics heralds a new era of exploration and comprehension. The implementation of computational methods for the preservation, inheritance, and promotion of Cultural Heritage has emerged as a prominent research trend worldwide since the 1990s. This trend reflects a growing recognition of the importance of utilizing digital tools and techniques to safeguard and transmit cultural heritage to future generations.

Cultural informatics (CI), also known as Cultural Computing (CC), Heritage Informatics (HI), and Heritage Computing (HC), is an interdisciplinary field that focuses on the application of information and communication technologies (ICT) to the study, preservation, management, and dissemination of cultural heritage. It employs the use of computational methods, digital tools, and information systems to document, analyze, interpret, and present cultural artefacts, traditions, and practices. This includes the digitization of cultural materials to create digital surrogates that can be accessed, studied and shared online, aiming to democratize access to cultural heritage resources, promote cultural diversity and understanding and facilitate

WILDRE-7 Workshop, pages 30–39

research, education, and public engagement with cultural heritage (Balakrishnan and Yogeshwaran, 2018). CI, with its focus on computational methods and technological tools, opens up fresh avenues for delving into the depths of the DS. CC specifically focuses on the practical application of computers and computational technologies in various aspects of cultural preservation, including recovery, storage, modeling, recreation, presentation, and communication (Tosa et al., 2005). Situated at the intersection of computer science, humanities, and cultural studies, CC attempts to analyze, interpret, preserve, and disseminate cultural artifacts, practices, and expressions through digital means.

HC, on the other hand, aims to enhance understanding of culture, facilitate cultural heritage preservation, and foster communication and engagement within and across cultural communities. In a country like India, characterized by a diverse landscape, multi-lingual populace and cultural intricacies, HC plays a crucial role in safeguarding and nurturing the nation's cultural heritage. By harnessing digital mediums and computer technologies, HC guarantees the preservation and accessibility of cultural treasures to individuals from all walks of life, promoting cultural exchange and communication (Meng, Wang and Xu, 2022), while transcending geographical and linguistic boundaries ensuring its endurance for posterity (Manjulaadevi and Geethalakshmi, 2019).

This paper outlines the methodology and technical framework of the concept mining system, elucidates the challenges encountered in analyzing age-old texts through computational means, and discusses the potential implications and applications of Dharmaśāstric informatics in the fields of cultural studies, history, anthropology, and beyond. Through interdisciplinary collaboration and innovative research methodologies, it is a pursuit to illuminate the socio-cultural facet of ancient India encapsulated within the timeless wisdom of the DS.

2. Sociocultural Dynamics in Dharmaśāstric Knowledge Traditions

DS encompasses a rich reservoir of knowledge pertaining to societal governance, ethical conduct, and cultural norms. Rooted in the principles of dharma, these texts offer profound insights into the socio-cultural dynamics prevalent in ancient Indian society. This section explores the intricate interplay between sociocultural dynamics and theological knowledge traditions, shedding light on their enduring relevance and impact.

To comprehend the sociocultural dynamics embedded within DS texts, it is essential to delve into the historical context of ancient India. During this period, society was structured hierarchically, with distinct varnas (castes) and ashramas (stages of life). Dharma, the moral and ethical duty prescribed for each varna and ashrama, formed the cornerstone of societal order and cohesion. DS, comprising texts such as *manusmṛti*[1] *yājñavalkyasmṛti*[2], *nāradasmṛti*, *arthaśāstra*, and *dharmasūtra*, provided guidelines for individuals and communities to uphold dharma in their respective roles and responsibilities.

2.1 *Varṇāśrama* or Gender Roles:

One of the key aspects of sociocultural dynamics elucidated in DS texts is the delineation of gender roles and family structure. These texts prescribe specific duties and obligations for men and women based on four main varnas (Chaubey, 2005) namely; *brāhmaṇa* (priests and scholars), *kṣatriya* (warriors and rulers), *vaiśya* (merchants and farmers), and *śūdra* (labourers and servants) and 4 segments of life known as ashramas; *brahmacarya* (student life), *gṛhastha* (householder life), *vānaprastha* (retired life), and *saṃnyāsa* (life of renunciation). Each stage has its own set of duties and obligations (Chander, 2015).

2.2 Vivāha or Marriage/Family Structure

While men were primarily responsible for providing sustenance and protection, women were entrusted with domestic duties and nurturing familial bonds. The institution of marriage was revered, serving as a cornerstone of societal stability and continuity. It delineates the rights and responsibilities of spouses, and provides guidelines for marriage, including rules for choosing a suitable partner, conducting marriage ceremonies, the concept of dowry and women's property, and elucidating the principles of mutual respect, fidelity, support within marital relationships and cardinal importance of a stable family unit.

[1]dhṛtiḥ kṣamā damo'steyaṃ śaucamindriyanigrahaḥ. dhīra vidyā satyamakrodho daśakaṃ dharmalakṣaṇam. manusmṛti 6.92

[2]ahiṃsā satyama'steyaṃ śaucamindriyanigrahaḥ. dānaṃ damo dayā śāntaḥ sarveṣāma dharmasādhanam. yājñavalkyasmṛti1.122

2.3 Caste System and Social Hierarchy

The caste system, a prominent feature of traditional Indian society, also finds mention in DS texts. These texts categorize individuals into varnas as previously mentioned, based on their inherent qualities and occupations, prescribing distinct rights and duties for each varna. While the varna system was intended to foster social order and cooperation, it also perpetuated hierarchical divisions and inequalities. DS underscored the importance of upholding varna dharma, thereby reinforcing social cohesion and stability.

2.4 Ethical Conduct and Justice

Ethical conduct and justice are integral components of DS knowledge traditions. These texts delineate principles of righteous conduct (dharma) and advocate for the equitable dispensation of justice. The concept of dharma encompasses moral, ethical, and legal obligations, guiding individuals in their interactions with others and society at large (Sankhder, 2003). DS also elucidate the principles of punishment and restitution, emphasizing the importance of upholding justice while mitigating harm.

2.5 Rājadharma Kingship or rule of state

DS offer insights into the responsibilities of kings and rulers. They outline the principles of just governance, administration of justice, and the welfare of the subjects (Nath, 2019).

3. Research Problem and Objective

Scholars globally have extensively examined DS texts, leading to a resurgence of interest in their traditional concepts and literary heritage. The wealth of knowledge preserved in Sanskrit has attracted scholars from India and the West, underscoring the importance of accessible Sanskrit resources for fostering widespread discourse on Sanskrit knowledge. Despite the ongoing scholarly engagement with DS texts, there is a growing imperative to explore their scientific nuances and technological perspectives. In today's globalized and digitally advanced era, characterized by widespread internet access and the proliferation of technological innovations, there is a notable surge in demand for online educational resources. However, the absence of an instant information retrieval system or online indexing apparatus specifically tailored for DS texts remains a critical gap. Despite efforts to digitize educational materials, Sanskrit texts remain largely inaccessible in electronic formats. Covering a broad spectrum of subjects, such as traditions, culture, history, and ancient scientific insights, these texts face challenges in accessibility, hindering extensive knowledge discourse and research in this field. Given the contemporary landscape of globalization and digital innovation, there is an urgent need for an instant information retrieval system or online indexing tool based on DS texts to enhance accessibility and facilitate further research and study in this area.

The primary objective of this research is to develop a Web-based Search Mechanism and IE Mechanism for DS texts, as well as, the implementation of a concept mining system tailored for exploring the socio-cultural facets embedded within the Indian society as depicted in ancient DS texts. This system aims to address the lack of accessibility to Sanskrit resources, by providing a user-friendly platform for scholars, Sanskritists, sociologists, experts in management sciences, political scientists, economists, legal experts, *āyurveda ācāryas*, and various science experts to access and study these texts thoroughly.

The developed system aims to delve into the rich repository of socio-cultural knowledge encapsulated within these texts, facilitating a deeper understanding of the societal dynamics, norms, and traditions prevalent during that era. By leveraging informatics methodologies, the research seeks to address the challenge of extracting and analyzing complex socio-cultural concepts from vast and intricate DS literature, ultimately contributing to scholarly discourse and knowledge dissemination in the field of ancient Indic studies, thus, promoting the global impact of Sanskrit literature in the field of world science.

4. Data Mining Techniques and Concurrent Surveys

Information or data mining is the process of extracting valuable information from large datasets, often through the analysis of data patterns or the use of predefined rules with software. It involves searching through extensive documents or unstructured text to extract relevant information, ideas, and content. Sanskrit, with its vast literary tradition, presents a rich source of such data, necessitating the development of online systems to access and extract specific information from texts, particularly within DS knowledge traditions.

Concept mining is akin to text and data mining but focuses on uncovering underlying ideas and topics within documents or unstructured text. It involves creating mining models and applying artificial intelligence to find intent and deep-rooted meaning (Feldman and Dagan 1955). It

focused on extracting target-based information from a corpus. It extracts accurate references or information even when the searched queries or input keywords are not directly or explicitly visible (Huet, 2005). Therefore, optimizing the veracity of the search results.

CL Techniques pivotal for Mining and Extracting Information from DS Texts, are as follows:

1. **Text Mining**: This mining technique can be employed to extract valuable insights and information from the vast corpus of DS literature. By analyzing the text computationally, researchers can identify patterns, trends, and recurring themes within the DS texts, shedding light on the principles of dharma, societal norms, legal frameworks, and ethical guidelines advocated in these texts.

2. **Natural Language Processing (NLP)**: NLP techniques are crucial for understanding the nuances of Sanskrit language used in DS texts. Techniques such as tokenization, part-of-speech tagging, and named entity recognition can aid in parsing the text, identifying key concepts, and annotating entities such as individuals, deities, locations, and legal terms mentioned in the DS texts.

3. **Information Retrieval (IR) or Extraction (IE)**: IR techniques are essential for efficiently retrieving relevant information from DS texts. By indexing the texts and implementing retrieval models, scholars can quickly locate specific passages, verses, or sections related to particular topics, allowing for focused study and analysis.

4. **Named Entity Recognition (NER)**: NER techniques are particularly useful for identifying named entities within DS texts. Scholars can use NER algorithms to automatically annotate names of sages, rulers, legal authorities, and other entities mentioned in the DS literature, facilitating the identification and analysis of key figures and references.

5. **Topic Modelling**: This technique enables researchers to uncover latent topics or themes present in DS texts. By applying algorithms such as Latent Dirichlet Allocation (LDA) or Non-negative Matrix Factorization (NMF), scholars can identify clusters of related concepts or discourses within the DS corpus, providing insights into the diverse subjects addressed in these texts.

6. **Sentiment Analysis**: Another technique that falls under the category of HC is sentiment analysis. While it may not directly apply to DS texts in the same way as modern textual data, analogous techniques can be employed to discern attitudes, emotions, and moral judgments expressed within the texts. By analyzing linguistic cues and contextual clues, researchers can gain a deeper understanding of the ethical and moral dimensions conveyed in DS literature.

In India, prominent institutes such as the School of Sanskrit and Indic Studies at Jawaharlal Nehru University, the Department of Sanskrit Studies at the University of Hyderabad, and the Department of Sanskrit at the University of Delhi are actively engaged in research and development related to computational Sanskrit. Their work primarily focuses on information mining and search techniques for Sanskrit texts, including online indexing and instant referencing systems for various texts such as the *Amarakośa, Mahābhārata, Nirukta, Vedāṃta*, and more. Notably, Jha (2006) has made significant contributions in the field of Sanskrit text summarization. One notable project is the Online Multilingual *Amarakośa* system, based on the ancient Sanskrit thesaurus *Amarakośa* attributed to *Amarasiṁha*. This system, developed using RDBMS techniques, allows users to search for up to 50 synonyms along with category, gender, number information, and detailed glosses. It enables cross-referencing among synonyms, supports search capabilities in various Indian languages, and offers an ontology display. Users can employ this system to search for any word found within the text of *Amarakośa* (Khandoliyan, 2011).

Similarly, efforts have been made to digitize and enable online search for *Purāṇas*, (Anju and Chandra, 2017), *Sāṃkhya*-yoga technical terms database (Anju and Chandra, 2018). Efforts have been undertaken to facilitate online search and indexing for various texts such as the *Mahābhārata* (Mani, 2010), Nirukta (Soni, 2009), *Medinīkośa* (Dwivedy, 2009), *Maṅkhakośa* (Kumar, 2009) etc. For instance, the *Āyurveda* Search system is based on the works of *Caraka saṃhitā* (Tiwari, 2011) and *Suśruta saṃhitā* (Pandey, 2011), enabling users to search for specific terms and concepts related to Ayurvedic texts. The *Vedānta* Search mechanism allows users to search for any word within *Vedānta* texts in the Vedas. Additionally, the *Ṛgvedika* Search system offers an instant search feature for the *Ṛgveda*, enabling users to search for mantras and words within the *Ṛgveda saṃhitā* at any time, providing immediate references when needed. The information from any of the *mantras*

of *Ṛgveda* can be searched in many ways such as deity, *maṇḍala*, *ṛṣī* etc (Kumar, 2016). One of the major works carried out in this field is the digitization of the heritage theological text *Manusmṛiti* (Nigam and Chandra, 2022), wherein the entire text is digitally indexed and technical terminologies and concepts are electronically mined using interactive search techniques. These steps encircle the digital representation and preservation of cultural heritage resources, documents, traditional knowledge, and intangible cultural practices.

5. Data Collection and Research Methodology

The methodology entails leveraging digital resources to explore how DS wisdom can be made more accessible and relevant in modern education. Through systematic data collection and analysis, the study aims to compile a comprehensive digital corpus of DS texts and related materials.

1. **Digital Resource Identification:** Conduct a systematic search across various digital platforms including libraries, databases, repositories, and websites to gather a comprehensive collection of digital resources related to DS knowledge.
2. **Data Collection:** Gather DS texts, interpretations, commentaries, and primary/secondary educational materials available in digital formats to establish a robust digital corpus for analysis. The included DS texts and their respective structures are outlined below:
    - *Apastamba Dharmasūtra*: 1,364 sutras
    - *Gautama Dharmasūtra*: 973 sutras
    - *Baudhāyana Dharmasūtra*: 1,236 sutras
    - *Vasishtha Dharmasūtra*: 1,038 sutras
    - *Yājñavalkyasmṛti*: 1,010 ślokas
    - *Nāradasmṛti*: Approximately 2000 verses
    - *Viṣṇusmṛti*: Approximately 2000 verses.
3. **Content Analysis:** Evaluate digitized materials by scrutinizing them for key themes, principles, and pedagogical elements to assess the quality and authenticity of different digital resources.
4. **System Development:** The aim is to extract sociocultural concepts from DS texts by combining Computational Linguistics and search methodologies. This involves developing a web-based system using Information Extraction (IE) methods, web technology, and CI for searching. To enhance search effectiveness, data mining techniques like concept mining and digital indexing will be employed. Additionally, original verses from prominent DS scriptures will be integrated, and keyword searching (Gibb, 1992) can be utilized. Different computational research modus operandi will be explored to mine technical terminologies and distinct concepts from DS texts, aiming to create an accurate and error-free system for deriving conceptual insights.

5.1 Digital Platform

The Instant Information Retrieval and Concept Mining System for DS texts is an online, web-based, input-output generating system. Utilizing a tagging technique, this system boosts its capacity to extract verses, even in cases where the directly queried word may not be present (Huet, 2005). By applying text mining, natural language processing, and semantic analysis techniques, our system aims to extract, categorize, and interpret key concepts about societal organization, ethical conduct, familial relationships, and religious practices. The system consists of two main components: the Front-End and the Back-End. The Front-End, developed using HTML, CSS, and JavaScript, provides the user interface. Meanwhile, the Back-End includes the programming logic, databases, and servers. Python serves as the programming language, with data stored in text files and Flask utilized as the server (Khandoliyan, Pandey, Tiwari, & Jha, 2012).

The following steps have been taken to for the development of the system:

1. Creation of a digital database containing all DS scriptures mentioned earlier, storing original *ślokas* and translations in separate UTF-8 Devanagari format text files organized within designated databases.
2. Compilation of a list of conceptual terms from prominent DS scriptures, along with their translated meanings.
3. Development of a database containing English and Hindi exegesis to provide detailed explanations and interpretations of these concepts.
4. Creation of a Script Validator Module to validate user query input scripts, distinguishing between Devanagari and IAST scripts for proper processing.
5. Establishment of a Concept Validator to match concept information based on user

search queries, ensuring accurate retrieval from the database.

Upon receiving the user's query, a coordinated effort involving multiple programs will be initiated to produce the desired output. The pre-processor, situated in the backend, will execute the initial query and synchronize it with the digital information indexer. Simultaneously, the script validator will confirm the input language, while the concept indexer will align relevant verse tags with the query. Additionally, the meaning generator will provide detailed explanations of the verses. Subsequently, the system will search subsequent queries from different databases, and the output generator will generate corresponding results, formatted according to the user's query input and displayed on the client's end.

An intuitive user interface facilitates user interaction and query submission. This system operates as a cohesive mechanism, leveraging various digital components to achieve its objectives. The key components include the User Interface, Preprocessor, Information Extractor, Information Generator, Meaning Generator, Concept Generator, Script Validator, and Output Generator. The system offers two input options and delivers analyzed output accordingly. The first option, termed "Direct Search," allows users to input any keyword in either Devanagari UTF-8 or Roman IAST format, receiving references, translations, and exegesis from the relevant DS manuscripts. The second option provides a "Dropdown Menu" feature, enabling users to select keywords from a pre-established list of concepts within the DS manuscripts, quickly accessing accurate information. Upon clicking on an indexed word, the system presents detailed information along with the corresponding *śloka* where it appears. The user interface efficiently processes user input and displays the output on the same page.

6. Features of the System

The developed system exhibits a high level of efficacy in responding to user queries, offering a seamless experience by accommodating inputs in both Devanagari and Roman scripts and presenting results in the chosen format. Leveraging advanced online indexing and tagging techniques, the system empowers users to explore any concept or word within DS texts. Key Features of the System:

1. **User-Friendly Interface:** The system's interface is designed for ease of use, allowing users to input queries effortlessly. It supports Keyword, Concept, and Phrase searching, enhancing flexibility for users seeking diverse information (Harter, 1975; Hulth et al., 2001).
2. **Bilingual Capabilities:** The system is capable of processing queries and producing results in both Devanagari and Roman scripts, promoting inclusivity for users with different language preferences.
3. **Comprehensive Output:** Search results include specific ślokas, complete references (book name, chapter number, verse number), and hyperlinks to meanings and explanations.

   Hovering over a śloka provides instant access to its meaning in Hindi and English while clicking on a verse retrieves automatic interpretations in both languages.
4. **Information Retrieval Module:** The system operates on a precise conceptual information retrieval module, ensuring accuracy and speed in delivering relevant information.

   Users benefit from quick and error-free retrieval of information related to DS concepts.
5. **Concept Tagging for Embedded Concepts:** In cases where DS concepts are embedded in MS verses without explicit mention of the query word, the system employs DS concept tagging.

   For example, the system successfully retrieves a verse related to the *varṇa* system, even if the word "*varṇa*" is not explicitly present.

7. Result and Discussions

Exploring DS Informatics delves into three key aspects: methodology, challenges, and potential applications. The proposed methodology involves the integration of computational techniques with traditional scholarly approaches to analyze DS texts. It employs text mining, natural language processing (NLP), and semantic analysis to extract, categorize, and interpret socio-cultural concepts embedded within these texts. By developing algorithms and tools tailored to the unique linguistic and thematic characteristics of *Dharmaśāstric* literature, it aims to uncover the multifaceted socio-cultural landscape of ancient India. The presented approach emphasizes the systematic exploration of textual data, enabling us to identify patterns, correlations, and underlying principles that shed light on the socio-cultural dynamics of the time.

Despite the promise of *Dharmaśāstric* Informatics, several challenges must be addressed. Firstly, the complexity and ambiguity of ancient texts pose significant hurdles for computational analysis. The nuanced language, metaphorical expressions, and cultural context inherent in *Dharmaśāstric* literature require sophisticated computational models capable of discerning subtle meanings and nuances. Additionally, the diversity of interpretations and commentaries on DS texts further complicates the analysis process. Furthermore, the scarcity of digitized and annotated texts presents challenges for training and validating computational models. Overcoming these challenges requires interdisciplinary collaboration, innovative algorithm development, and careful validation against traditional scholarly interpretations.

Despite these challenges, Dharmaśāstric Informatics holds immense potential for advancing our understanding of ancient Indian society and culture. By elucidating the socio-cultural landscape reflected in Dharmaśāstric texts, our approach can contribute to various fields of study, including history, anthropology, sociology, and religious studies. The insights gleaned from DS Informatics can inform contemporary discourse on issues such as governance, ethics, family structure, and religious practices. Furthermore, our methodology can facilitate comparative studies across different DS texts and commentaries, enabling a deeper exploration of regional variations and historical developments. Additionally, Informatics has practical applications in heritage preservation, education, and cultural revitalization efforts, ensuring that the wisdom of ancient India continues to enrich contemporary society.

The system as discussed above has been developed by the Computational Linguistics Research & Development, Department of Sanskrit, at the University of Delhi and can be accessed at https://cl.sanskrit.du.ac.in. It marks a significant advancement in facilitating user-friendly access to DS texts. With its ability to accept inputs in both Devanagari and Roman (IAST) scripts, the system ensures that outputs are generated in the corresponding script, enhancing accessibility for users. Leveraging information mining, online indexing, and tagging techniques, the system enables effortless searching of DS concepts, disparate *ślokas*, and words within manuscripts. Currently, it is working for Manusmṛti texts but in the future prominent DS texts will be added.

The system's functionality encompasses keyword, concept, and phrasal searching through online indexing modules, providing comprehensive information for each query. Results include the original *ślokas* with accurate references, indicating the chapter and verse numbers for easy reference. Moreover, each verse is hyperlinked, allowing users to access word meanings and complete exegesis. By hovering over a *śloka*, users can view bilingual explanations, and clicking on it provides automatic interpretation in both Hindi and English.

This system's capability to deliver complete information on any concept, including original *ślokas*, bilingual translations, and interpretations, underscores its utility and potential impact in facilitating research and study of DS texts.

Conclusion and Future Directions of Research

In summary, DS Informatics offers a novel approach to uncovering the socio-cultural landscape of ancient India through computational analysis of Dharmaśāstric texts. While challenges exist, the potential applications of this approach are far-reaching, promising to shed new light on India's rich cultural heritage and inform contemporary discourse on socio-cultural issues. By harnessing the power of CI, scholars can employ advanced algorithms and analytical techniques to unravel the intricacies of these ancient texts in ways that were previously unimaginable. Computational methods such as text mining, natural language processing, and semantic analysis offer the promise of uncovering hidden patterns, correlations, and insights buried within the DS.

Moreover, informatics enables scholars to explore the interconnections between different sections of the texts, discerning overarching themes and recurrent motifs that provide a deeper understanding of ancient Indian society. By leveraging computational tools, researchers can conduct large-scale analyses across multiple Dharmaśāstric texts, facilitating comparative studies and highlighting regional variations and historical developments.

In essence, while traditional methods of textual analysis remain invaluable, the integration of informatics into the study of the DS opens up new vistas of exploration and understanding. By marrying ancient wisdom with modern technology, scholars can illuminate the socio-cultural fabric of ancient India with unprecedented depth and clarity, enriching our appreciation of this profound cultural heritage.

In India's rich cultural heritage, the emergence of HC and Digital Heritage stands as a pressing need in the contemporary era. As India embarks on its "Digital India" campaign, the goal is to ensure that every citizen has access to and

proficiency in utilizing digital mediums, thereby placing the nation on equal footing with developed countries. However, amidst this digital transformation, it is crucial to recognize the significance of preserving and leveraging India's cultural heritage in the digital realm. Heritage Computing and Digital Heritage initiatives play a pivotal role in this enterprise by digitizing, cataloguing, and disseminating India's vast cultural legacy through digital platforms. By harnessing technology, these efforts not only facilitate broader access to India's rich heritage but also contribute to its preservation and promotion on a global scale (Manjulaadevi and Geethalakshmi, 2019). In essence, Cultural Computing and Digital Heritage initiatives ensure that the digital revolution encompasses not just technological advancement but also the preservation and celebration of India's cultural identity. The future directives for this system can be implored as discussed:

1. **Cross Reference:** Cross-referencing allows researchers to validate their findings by comparing them with those from other sources. By corroborating information across multiple references, researchers can enhance the credibility and reliability of their research outcomes. It also facilitates the identification of patterns, trends, or commonalities in the interpretation or usage of specific terms or concepts. It helps in placing the terms or concepts within their broader context. By exploring how these terms are used or understood in different cultural, historical, or disciplinary contexts, researchers can gain deeper insights into their meanings and implications.
2. **Cross-Linguistic Analysis:** Conducting cross-linguistic analysis using computational methods can facilitate comparative studies between classical Indian texts and texts from other linguistic traditions, fostering interdisciplinary research and enriching our understanding of linguistic and cultural exchange.
3. **Multimodal Analysis:** Integrating multimodal analysis techniques that combine textual data with images, audio recordings, and other multimedia elements can provide a more holistic view of classical texts, enhancing their interpretability and engaging users in immersive learning experiences.
4. **Enhanced System Functionality:** Continuously improve the user interface and system functionality based on user feedback and emerging technologies. This could involve incorporating advanced search algorithms, expanding the database of DS texts, and refining the accuracy of information retrieval.
5. **Collaboration and Partnerships:** Foster collaborations with academic institutions, research organizations, and cultural heritage institutions to expand the scope of the research and access additional resources. Collaborative efforts can lead to the discovery of new DS texts, improved data collection methodologies, and broader dissemination of research findings.
6. **Multilingual Support:** Extend and enhance the system's multilingual capabilities to support digitization efforts across a wide range of Indian languages, including but not limited to Sanskrit, Tamil, Telugu, Kannada, Bengali, Urdu, Marathi etc. enabling users from diverse linguistic backgrounds to access DS texts and resources. This involves incorporating translation services, language-specific lexicons, grammars and linguistic resources, multilingual interfaces, and expanding the database to include texts in other languages.
7. **Digitizing other Classical and Heritage Texts:** The developed model can be further expanded, modified and appropriately applied for digitizing the classical texts as well as heritage texts across all Indian languages, presenting a promising future direction with immense scholarly and cultural significance. By leveraging the model's robust framework and adapting it to the diverse linguistics literature texts such as:

   7.1 Language Adaptation: Modify the model to accommodate the unique linguistic features, scripts, and writing systems of various Indian languages. This involves developing language-specific modules for text processing, analysis, and representation to ensure accurate digitization and preservation of classical and heritage texts.

   7.2 Collaborative Partnerships: Foster collaborations with linguistic experts, historians, archaeologists, librarians, and cultural institutions across India to access and digitize a diverse range of classical and heritage texts. By leveraging domain expertise and resources for text-specific data collection, collaborative efforts can accelerate digitization initiatives and ensure comprehensive coverage of Indian literary traditions.

   7.3 Community Engagement: Engage with local communities, scholars, students, and enthusiasts to crowdsource content, gather annotations, and validate digitized texts. By

involving stakeholders in the digitization process, the model can benefit from collective knowledge and ensure the relevance and utility of digitized materials for diverse user groups.

8. **Education and Outreach:** Conduct workshops, training programs, and outreach activities to raise awareness about the importance of DS texts and the potential applications of the research findings. Engage with educators, students, and the general public to promote the use of digital resources for studying DS knowledge traditions.

9. **Interdisciplinary Research:** Encourage interdisciplinary research collaborations to explore the intersection of DS knowledge with other fields such as linguistics, anthropology, philosophy, and computer science. Interdisciplinary approaches can lead to new insights and perspectives on DS texts and their cultural significance.
Castor, A. and Pollux, L. E. (1992). The use of user modeling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

Acknowledgment

This publication is the result of research supported by the Faculty Research Programme (FRP) from the Institution of Eminence (IoE), University of Delhi, Delhi with Ref. No./IoE/2023-24/12/FRP dated August 31, 2023. The authors gratefully acknowledge the use of the services and facilities of Computational Linguistics Research & Development, Department of Sanskrit, at the University of Delhi.

Bibliographical References

Anju, and Chandra, S. (2017). Puranic Search: An Instant Search System for Puranas. *Language in India.*

Anju, and Chandra, S. (2018). Sāṃkhya-yoga darśana paribhāṣā ḍeṭābesa evaṃ Onalāina khoja. *Research Review International Journal of Multidisciplinary,* 3(11):890-894.

Balakrishnan, S. and Yogeshwaran, R. (2018). Heritage Computing and its Impact. *Computer Society of India Communications*. 42(10):6-7.

Banerji, S. C. (1999). A Brief History of Dharmaśāstra. Abhinav Publications.

Biswas, S. and Banerjee, D. (2016). The Dead Language Sanskrit is not actually dead. *Journal of Education and Development*, 6(12):90-97.

Chander, R. (2015). Arthśastra: A Replica of Social Dynamism in Ancient India. *International Journal of Innovative Research and Advanced Studies*, 2(4):78-83.

Chaubey, S. (2005). Vedoṃ meṃ dharma kī avadhāraṇā. *Doctorate thesis*. Faizabad, India: Dr Rāma Manohara Lohiyā Avadha Viśvavidyālaya.

Dubey, V. K. (2012). Vedang Shiksha Sahitya me Vyasshhhiksha ek Parisheelan. *Doctorate thesis*. Faizabad, Uttar Pradesh, India: Dr. Rammanohar Lohia Avadh University.

Dwivedy, P. K. (2009). Medinikosh Project. *M.Phil dissertation*. New Delhi, India: Special Center for Sanskrit Studies, J.N.U.

Feldman, R., *and* Dagan, I. (1995). Knowledge Discovery in Textual Databases (KDT). KDD, 95, pp. 112-117.

Gibb, F. (1992). Knowledge-based indexing. The Application of Expert Systems in Libraries and Information Centres, pp. 34-67.

Harter, S. P. (1975). A Probabilistic Approach to Automatic Keyword Indexing, Part II, An algorithm for probabilistic indexing. *Journal of the American Society*, 26(5):280-289.

Huet, G. (2005). A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *Journal of Functional Programming*, 15(4):573-614.

Hulth, A., Karlgren, J., Jonsson, A., Boström, H., and Asker, L. (2001). Automatic keyword extraction using domain knowledge. In International Conference on Intelligent Text Processing and Computational Linguistics pp. 472-482. Berlin: Springer.

Jha, G. N. (2006). Computational lexicography and Amarakosha: an online RDBMS approach. *National Seminar of Language and Interface.* Deparment of Linguistics. University of Delhi.

Khandoliyan, B. R. (2011). *Vanaushaadhi-varga of Amarakosha: A computational study*. Special Center for Sanskrit Studies, Jawaharlal Nehru University, New Delhi, India.

Khandoliyan, B. R., Pandey, R. K., Tiwari, A., and Jha, G. N. (2012). In P. Osenova, S. Piperidis, M. Slavcheva, & C. Vertan (Eds.), Text encoding and search for Āyurvedic texts: An interconnected lexical database. *Adaptation of Language Resources and Tools for Processing Cultural Heritage Objects*, *2*:36-42.

Kumar, A. (2009). *Maṃkha-Kośa Project*. New Delhi, India: Special Center for Sanskrit Studies, J.N.U.

Kumar, J. (2016). *M.Phil dissertation*. New Delhi, India: University of Delhi.

Mani, D. (2010). *RDBMS Based Lexical Resource for Indian Heritage: The Case of Mahābhārata*. Presented at the International Sanskrit Computational Linguistics Symposium. Berlin, Heidelberg.

Manjulaadevi, K. & Geethalakshmi N. (2019). Impact of Digital Heritage and Heritage Computing. Asian Journal of Computer Science and Technology ISSN: 2249-0701, 8 (S1):25-27.

Meng, Li, Wang, Yun & Xu, Yingqing. (2022). Computing for Chinese Cultural Heritage. Visual Informatics 6(1):1-13.

Nath, R. (2019). Good Governance and Ancient Indian Administration. Bihar Journal of Public Administration, 276.

Nigam, A., & Chandra, S. (2022). Digital World of Dharmaśāstric Knowledge Tradition: An Instant Information Retrieval System for Manusmṛiti. *GIS: Science Journal, 9*(8):241-249.

Pandey, R. K. (2011). Online Indexing Of Sushruta Samhita. *D*octorate *diss*ertation. New Delhi: Special Centre for Sanskrit Studies, Jawaharlal Nehru University.

Phillips, Stephen H. 2014. Epistemology in classical India: The knowledge sources of the Nyaya school. UK: Routledge.

Sankhder, M. M. (2003). Democratic Politics and Governance in India. Deep and Deep Publications.

Soni, C. (2009). Niruktanirvacana Project. New Delhi, India: School of Sanskrit and Indic Studies.

Tiwari, A. (2011). Online Indexing in Caraka Samhita. *M.Phil Dissertation*. New Delhi: Special Centre for Sanskrit Studies, JNU.

Tosa, N., Matsuoka, S., Ellis, B., Ueda, H., Nakatsu, R. (2005). Cultural Computing with Context-Aware Application: ZENetic Computer. In: Kishino, F., Kitamura, Y., Kato, H., Nagata, N. (eds) Entertainment Computing - ICEC 2005. ICEC 2005. Lecture Notes in Computer Science, vol 3711. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11558651_2

Exploring News Summarization and Enrichment in a Highly Resource-Scarce Indian Language: A Case Study of Mizo

Abhinaba Bala, Ashok Urlana, Rahul Mishra, Parameswari Krishnamurthy
International Institute of Information Technology - Hyderabad
{abhinaba.bala, ashok.urlana, rahul.mishra, param.krishna}@iiit.ac.in

Abstract

Obtaining sufficient information in one's mother tongue is crucial for satisfying the information needs of the users. While high-resource languages have abundant online resources, the situation is less than ideal for very low-resource languages. Moreover, the insufficient reporting of vital national and international events continues to be a worry, especially in languages with scarce resources, like **Mizo**. In this paper, we conduct a study to investigate the effectiveness of a simple methodology designed to generate a holistic summary for Mizo news articles, which leverages English-language news to supplement and enhance the information related to the corresponding news events. Furthermore, we make available 500 Mizo news articles and corresponding enriched holistic summaries. Human evaluation confirms that our approach significantly enhances the information coverage of Mizo news articles. The mizo dataset and code can be accessed at https://github.com/barvin04/mizo_enrichment .

Keywords: Low Resource Languages, News Enrichment, Mizo

1. Introduction

Low-resource languages often lack the required data resources for natural language processing tasks, hindering their inclusion in various applications. Significant progress has been made in generating open-source data for several scheduled Indian languages. However, languages like **Mizo** face persistent challenges in accessing domain-specific information. Mizo, a prominent member of the Tibeto-Burman language family, is primarily spoken by the Mizo people in India's northeastern region, especially in Mizoram, with significant populations in Manipur, Tripura, and Meghalaya. Additionally, it is also spoken in some parts of Myanmar and Bangladesh, further contributing to its linguistic diversity in the South Asian region. According to the 2011 census, the Mizo language had around 840,000 native speakers[1]. Mizo uses the Roman alphabet for its script.

Despite the presence of numerous newspapers in Mizo, limited NLP research has been conducted on this language. It is important to note that the sheer existence of numerous newspapers does not necessarily translate into an abundance of resources suitable for training NLP models. The scarcity of data in such languages remains a significant obstacle to performing essential NLP tasks, despite the progress made in this field for other languages.

The process of enriching articles written in low-resource languages through the utilization of auxiliary information represents an important advancement in the field of natural language processing.

[1] https://censusindia.gov.in/census.website/

Mizo text (truncated): Nimin khan Saron Veng, Serchhip district atangin chhungkaw 7 chu sawnchhuah an ni a, nimin khan ruahsur nasa avangin... nimin khan sawnchhuah an ni National Highway 54...occurred last year Chhungkaw pariat an awm tawh an chenna in aṭanga chhuahtiran ni. **En_translated:** Seven families from Saron Veng, Serchhip district were evacuated yesterday and today due to heavy rains ...were evacuated today National Highway 54...occurred last year Eight families have been evacuated from their homes.
En_enriched (truncated) : Eight people have been killed and six are missing after flash floods caused by heavy rainfall wrecked havoc in Tlabung in Mizoram's Lunglei district.... 350 houses have been submerged since yesterday. **Mizo_translated:** Mizoram's Lunglei district-a Tlabung khuaah ruahsur nasa vanga tuilianin a tihchhiat avangin mi 8 an thi tawh a, mi paruk chin hriat lohin an awm tawh a.... Nimin aṭang khan in 350 tuiin a chim tawh a ni.

Table 1: Example of the **(Top)** raw Mizo news article and corresponding translated version. **(Bottom)** corresponding enriched version of the same. Highlighted in **magenta** indicates the enrichment part, whereas **blue** signifies the context of the original article.

This auxiliary information serves as a repository of more pertinent and coherent information related to the original Mizo text as shown in Table 1. By incorporating auxiliary information, which could include translations, named entity recognition, summarization, information extractions, transcriptions, or contextual data from more widely studied languages, the Mizo articles gain depth, clarity, and broader

WILDRE-7 Workshop, pages 40–46

accessibility. This approach not only enhances the overall quality of content but also contributes to the continued documentation and dissemination of languages that might otherwise face the risk of being marginalized or lost over time. The utilization of auxiliary information exemplifies the intersection of technology and language conservation, fostering a bridge between underrepresented languages and the digital age while reinforcing the importance of linguistic diversity.

Our pipeline does not assume that events covered in Mizo and English news media are almost parallel. Instead, it aims to enrich the original Mizo articles with additional information available in English when feasible. The goal is to supplement the content, but we acknowledge that English may not always contain extra information on the specific events covered in the Mizo dataset.

In this study, we aim to enrich articles in low-resource languages by supplementing them with relevant information extracted from high-resource languages, such as English, using state-of-the-art NLP techniques. We introduce a straightforward pipeline that includes the following steps:

- Translate Mizo news article into English and generating a headline using state-of-the-art headline generation models.
- Extract valid URLs by querying the generated headline in a web search.
- Retrieve documents from the identified URLs and perform the multi-document summarization using state-of-the-art pre-trained models.
- Add the obtained summary to the corresponding document and translate the entire English document back to the Mizo language.

We have released the 500 Mizo documents and their corresponding enriched versions to facilitate further research on low-resource languages. To assess the pipeline's performance, we conducted a human evaluation. The results of the human evaluation indicate that the proposed pipeline effectively enriches low-resource language news articles.

2. Related Work

2.1. Mizo Datasets

Comprehensive datasets for Mizo language tasks are scarce, with a predominant focus on fundamental language understanding and translation rather than the creation of holistic summaries of Mizo news articles. Notably, research efforts such as Khenglawt et al. (2022) aim to address the scarcity of multimodal datasets for low-resource language pairs like English-Mizo. They present the Mizo Visual Genome 1.0 (MVG 1.0) dataset, featuring bilingual textual descriptions alongside images, facilitating English-Mizo multimodal machine translation. Additionally, Lalrempuii (2023) contributes an LUS dataset, a collection of 101,827 monolingual Mizo language sentences sourced from various news websites.

Description	Count
Mizo (single news) Documents	983
Mizo translated to English	983
Headlines	798
Articles with (valid + invalid) URLs	797
Articles without URLs	30
Articles with valid URLs (≥ 1)	767
Articles with valid URLs (≥ 2)	746
Total URLs	4054
Average URLs per document	**5.29**

Table 2: Mizo data statistics

2.2. Headline Generation

Generating headlines (Zhou and Hovy, 2004; Alotaiby, 2011; Panthaplackel et al., 2022) from articles simplifies information access and exploration. These succinct headlines can serve as search queries, aiding users in finding more related articles efficiently (Qumsiyeh and Ng, 2016).

2.3. Multi Document Summarization

Multi-document Summarization (MDS) involves the generation of a brief and condensed summary that includes the essential information from a collection of interconnected documents. Recent studies in MDS have demonstrated promise in both extractive (Angelidis and Lapata, 2018; Narayan et al., 2018) and abstractive (Chu and Liu, 2018; Fabbri et al., 2019; Liu and Lapata, 2019) summarization techniques.

3. Methodology

The methodology of this work leverages a simple pipeline that leverages state-of-the-art natural language processing (NLP) techniques to enrich articles in low-resource languages, such as Mizo, through the incorporation of auxiliary information from high-resource languages like English. The overarching process can be delineated into several key stages as shown in Figure 1.

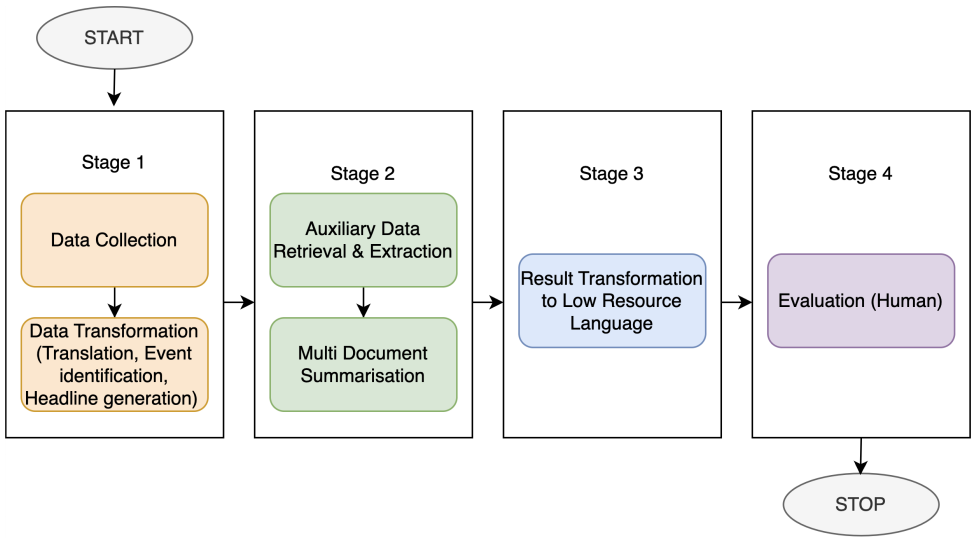


Figure 1: The Enrichment Methodology Pipeline. This illustration outlines the sequential stages of the methodology, which encompasses **a.** data collection, **b.** preprocessing, transformation/translation, **c.** headline generation, **d.** multi-document summarization, and **e.** translation into the low-resource language. These stages collectively contribute to the enrichment of articles in low-resource languages, facilitating a comprehensive understanding and accessibility of the content.

3.1. Data acquisition and preprocessing

To obtain Mizo raw data, we scraped publicly available information exclusively from the Mizoarchive[2], an online news portal. To ensure the creation of a high-quality dataset, we subject the collected data to necessary rule-based preprocessing steps. This involves the elimination of HTML tags and the removal of noisy text elements to preserve the integrity and quality of the source documents.

3.2. Data transformation

The data transformation includes translation from Mizo to English and obtaining the headline from the corresponding Mizo documents.

Translation from Mizo-English: Due to the absence of a Mizo summarization model, the initial step in our pipeline involves translating the cleaned Mizo document into English. We have utilized the Google-translate API to obtain Mizo to English translation. For the upcoming stages in the pipeline we have utilized English translated Mizo document.

Headline generation We employed state-of-the-art headline generation models to create headlines from English-translated Mizo documents. Specifically, we use the BART-large model (Lewis et al., 2020) fine-tuned on the CNN dataset for the headline generation task.

[2] https://mizoarchive.wordpress.com/

3.3. Information extraction

Obtaining valid URLs: Upon querying the headline in a web search, we retrieved various URLs. A URL was deemed valid if it directed to a Mizo news-article. We excluded URLs from major platforms like Wikipedia and YouTube. Comprehensive details on the criteria defining a valid URL are available in Table 2. Our approach encompasses documents of varying lengths, and we consider all topics without selective exclusions, ensuring a thorough exploration. On average, we obtained 5.29 valid URLs for each query.

Information retrieval: To acquire pertinent information from each web page linked to valid URLs, we employed a web scraping technique to transform unstructured web data into a structured format. This process involved utilizing the "Google" search engine and Python libraries like BeautifulSoup and urllib2. Specifically, urllib2 was used for URL retrieval, while BeautifulSoup was employed for data extraction.

In the pursuit of data quality and relevance, we meticulously selected the most contextually relevant URLs for the query. We also extracted valuable meta information from HTML tags like headings, paragraphs, tables, and images. Any incomplete sentences or irrelevant headings were intentionally excluded. After careful consideration of multiple sources, meaningful sentences were extracted and consolidated into a single document.

3.4. Uni-document Summarization

After the information extraction step, for each En-Mizo document, we have more than one relevant document. The assumption would be each document covers the relevant information with respect to the En-Mizo document. In this step, we have utilized the PEGASUS (Zhang et al., 2020) large model to generate individual summaries for each document. Subsequently, we concatenated all these summaries and fed the concatenated result back into the PEGASUS model to produce a coherent summary of all the documents.

3.5. Enrichment of Low Resource Language Articles

The final step of this pipeline is to translate all the English summaries into Mizo. This step ensures the conversion of the outcome from the high-resource language(s) into the target low-resource language. This step is pivotal in ensuring that the conclusions, findings, and insights derived from the analysis are made accessible and comprehensible to users who primarily operate within the context of the low-resource language. The next section validates the quality of the corresponding summary by performing the human evaluation.

4. Human Evaluation

4.1. Guidelines

To assess the quality of the pertinent information acquired through the proposed methodology, we conducted human evaluation. We randomly selected 50 documents from our meticulously curated Mizo dataset and engaged a native and proficient Mizo speaker to evaluate the generated summaries. We have provided the original Mizo document and the obtained summaries from the proposed pipeline and instructed the evaluator to assess the quality based on the following four distinct categories:

- Coherency: Assessing the logical flow and consistency of the summaries.
- Enrichment: Evaluating how effectively the summaries enhanced the original content.
- Relevancy: Determining the degree of relevance of the summaries to the original documents.
- Readability: Gauging the ease with which the summaries could be comprehended.

Each category was assessed on a scale from 0 to 4, with 0 indicating very poor, 1 representing somewhat unfaithful, 2 denoting moderate, 3 indicating good, and 4 representing near-perfect performance.

Coherency	Enrichment	Relevancy	Readability
3.82	2.44	2.9	3.98

Table 3: Human evaluation results

4.2. Analysis and discussion

- Coherency: The evaluation resulted in a relatively high level of coherency (3.82), suggesting that the logical flow and consistency of the summaries are generally well-maintained, contributing to their overall quality.
- Enrichment: The enrichment category received a score of 2.44, indicating moderate effectiveness in enhancing the original content within the generated summaries. The summaries appear to contribute to enriching the original content to a reasonable extent, but there are opportunities for refinement to make them more effective in this regard.
- Relevancy: The obtained score for relevancy is 2.9, suggesting a moderately relevant connection between the summaries and the original documents. This score indicates that the summaries exhibit a degree of alignment with the source documents, providing a basis for understanding the content.
- Readability: The readability score was relatively high at 3.98, indicating that the summaries are generally easy to comprehend. This is a positive aspect, as it ensures that the information can be accessible to a broader audience.

While coherency and readability seem to be relatively strong points, there is room for improvement in terms of enrichment and relevancy. These findings can guide future refinements of the summary generation pipeline, with the aim of achieving more comprehensive and contextually relevant summaries that enhance the original content to a greater extent.

5. Conclusion

In this work, we introduced a simple pipeline for enhancing low-resource (Mizo) news articles by infusing them with contextually relevant information. Our approach significantly improves the coverage of pertinent topics within Mizo documents, which is apparent from the results of our human evaluation. Additionally, this pipeline can be utilized to

boost news content in other underrepresented languages with only minor modifications to the overall approach.

6. Limitations

The effectiveness of the proposed methodology relies on the availability and relevance of auxiliary information from high-resource languages. In scenarios where such information is sparse or not applicable, the enrichment process may be hindered. The assumption that events covered in Mizo and English news media are parallel may not always hold true. Variations in news coverage and the uniqueness of local events may challenge the assumption that English can consistently supplement Mizo articles.

The chosen evaluation metrics, while providing valuable insights, might have limitations in fully capturing the nuanced aspects of enriching low-resource language articles. Further refinement and exploration of evaluation methodologies could enhance the robustness of the assessments. Human evaluation, while insightful, is inherently subjective. The interpretation of coherency, enrichment, relevancy, and readability can vary among evaluators, introducing a level of subjectivity that might impact the reliability of the assessments.

7. Ethics Statement

In conducting this research, we have prioritized key ethical considerations to uphold the integrity and responsibility of our work. Data privacy and informed consent are important, particularly when involving human subjects, ensuring that personal information is treated confidentially. Transparency is maintained through clear disclosure of data sources, methodologies, and any limitations present in the study. Cultural sensitivity is observed, avoiding misrepresentation and respecting the diversity of communities involved. Embracing open science practices, we aim to share code, datasets, and findings openly to foster collaboration and reproducibility.

Acknowledgements

We would like to express our sincere gratitude to the Mizo annotator(s).

References

Fahad T. Alotaiby. 2011. Automatic headline generation using character cross-correlation. In *Annual Meeting of the Association for Computational Linguistics*.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Eric Chu and Peter J. Liu. 2018. Meansum: A neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Vanlalmuansangi Khenglawt, Sahinur Rahman Laskar, Riyanka Manna, Partha Pakray, and Ajoy Kumar Khan. 2022. Mizo visual genome 1.0 : A dataset for english-mizo multimodal neural machine translation. In *2022 IEEE Silchar Subsection Conference (SILCON)*, pages 1–6.

Candy Lalrempuii. 2023. Lus: Mizo monolingual corpus.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. 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 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Sheena Panthaplackel, Adrian Benton, and Mark Dredze. 2022. Updated headline generation: Creating updated summaries for evolving news stories. In *Association for Computational Linguistics*, pages 6438–6461.

Rani Qumsiyeh and Yiu-kai Ng. 2016. Searching web documents using a summarization approach. *International Journal of Web Information Systems*, 12:83–101.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Liang Zhou and Eduard Hovy. 2004. Template-filtered headline summarization. In *Text Summarization Branches Out*, pages 56–60, Barcelona, Spain. Association for Computational Linguistics.

A. Appendix : Examples

As shown in Table 4, the enriched summary obtain high scores (all 4) by human evaluation. Where the context of *Turkey and US* is carried over as 'The two NATO allies'. The additional enrichment by our pipeline adds in the information about *equipment related to F-35 flighter aircraft.*

Table 5 shows the text for an example with scores (4, 2, 3, 4) for coherency, enrichment, relevancy and readability respectively. While there is moderate enrichment, there are parts which are not relevant or enriching enough according to the annotator.

Mizo text (truncated): Turkey President Recep Tayyip Erdogan chuan US-in Patriot missile a pawmpui chung pawhin Turkey chuan Russian S-400 missile lei tumna chu a thulh dawn lo tih a sawi. ... **En_translated:** Turkish President Recep Tayyip Erdogan has said that Turkey will not cancel its plan to buy Russian S-400 missiles despite the US approval of Patriot missiles. ...
En_enriched (truncated) : ... The two NATO allies have sparred publicly for months over Turkey's order for Russia's S-400 air defense system, which Washington says poses a threat to the Lockhead Martin Corp F-35 stealthy fighters, which Turkey also plans to buy. The United States has halted delivery of equipment related to the stealthy F-35 fighter aircraft ... **Mizo_translated:**... NATO tangrual ram pahnihte hi Turkey-in Russia-a S-400 air defense system a order chungchangah thla tam tak chhung vantlang hmaah an inhnial tawh a, Washington chuan Lockheed Martin Corp F-35 stealthy fighter-te tan hlauhawn tak a nih thu a sawi a, Turkey pawhin lei a tum bawk. US chuan F-35 fighter aircraft rukbo nena inzawm hmanrua pekchhuah chu a titawp ...

Table 4: Appendix-1, example of the **(Top)** raw Mizo news article and corresponding translated version. **(Bottom)** corresponding enriched version of the same. Highlighted in **magenta** indicates the enrichment part, whereas **blue** signifies the context of the original article.

Mizo text (truncated): ... "Kan zinkawngah hian MNF sorkarin hun rei lote chhungin hma a sawn a. Chief Minister Pu Zoramthanga chuan sorkar tharin hma a la dawn tih sawiin, conduct rules hian hun rei tak chhung min phuar tawh a ni" a ti a. **En_translated:** ... "MNF government has made progress in our journey within a short period of time. Chief Minister Pu Zoramthanga said that the new government is about to take action and the conduct rules have been binding us for a long time" ...
En_enriched (truncated) : Mizoram Chief Minister Zoramthanga on Sunday asserted that his party, the Mizo National Front, will return to power and bag 25-35 seats ... Mizoram CM Zoramthanga Reacting to allegations that the party is afraid of the BJP, the CM said the BJP-led central government ... **Mizo_translated:** Mizoram Chief Minister Zoramthanga chuan Pathianni khan a party, Mizo National Front chu thuneihna chang lehin seat 25-35 an la dawn tih a nemnghet a ... Mizoram CM Zoramthanga BJP an hlauhthawn nia puhna chhangin, CM chuan BJP kaihhruai central chu a sawi sawrkar ...

Table 5: Appendix-2, example of the **(Top)** raw Mizo news article and corresponding translated version. **(Bottom)** corresponding enriched version of the same. Highlighted in **magenta** indicates the enrichment part, whereas **blue** signifies the context of the original article.

Finding the Causality of an Event in News Articles

Pattabhi RK Rao, Sobha Lalitha Devi
AU-KBC Research Centre
MIT Campus of Anna University, Chennai-600044
sobhanair@yahoo.com

Abstract

This paper discusses about the finding of causality of an event in newspaper articles. The analysis of causality , otherwise known as cause and effect is crucial for building efficient Natural Language Understanding (NLU) supported AI systems such as Event tracking and it is considered as a complex semantic relation under discourse theory. A cause-effect relation consists of a linguistic marker and its two arguments. The arguments are semantic arguments where the cause is the first argument (Arg1) and the effect is the second argument(Arg2). In this work we have considered the causal relations in Tamil Newspaper articles. The analysis of causal constructions, the causal markers and their syntactic relation lead to the identification of different features for developing the language model using RBMs (Restricted Boltzmann Machine). The experiments we performed have given encouraging results. The Cause-Effect system developed is used in a mobile App for Event profiling called "Nigalazhvi" where the cause and effect of an event is identified and given to the user.

Key words: Causality extraction• Explicit intra-sentential causality • Implicit causality • Inter-sentential causality• Cause- effect, Event extractionThis work presents an automatic identification of explicit connectives and its arguments using supervised method, Conditional Random Fields (CRFs).

1. Introduction

In the last three decades researchers have successfully proved how to extract facts from unstructured text and also have developed large repositories which are focusing on is-a (Hearst, 1992) and part-of (Girju et al., 2003) relations. Information Extraction has many task which extracts facts such as Named Entity Recognition (NE), Relation Extraction (RE) and Event Extraction (EE). Cause–effect extraction is a relational extraction, a challenging task which requires semantic understanding and contextual knowledge of the unstructured text. Cause-effect relations appear frequently in any text. Consider the example which contains a cause-effect relation "heavy rain inundated the city." In this sentence "heavy rain" is the cause and the effect is "inundation of the city". A traditional definition of Cause-Effect relation can be as follows: An Event or Events that come first and results in the existence of another Event, ie, whenever the first event (the Cause) happens, the second event (the Effect) essentially or certainly follows.

The published work in this area can be classified into three approaches : knowledge-based, statistical/ML based , and deep-learning-based. And each method has its advantages and weaknesses. The knowledge-based approach uses linguistic patterns by using pre-defining hand-crafted rules or keywords (Garcia et al., 1997; Khoo et al., 2000; Radinsky et al., 2012; Girju et al., 2009; Kang et al., 2014; Bui et al., 2010). Statistical approach uses probabilistic models over features extracted (Girju, 2003; Do et al., 2011). Using CRFs cause-effect arguments were identified in (Menaka. S, et al., 2011). (Sindhuja G and Lalitha Devi, S 2017) where they consider the identification of causal relations across clauses and sentences using discourse connectives. This approach was applied on BIONLP/NLPBA corpus and identified the causal relations and causal entities. The most frequently used deep learning approaches are feed-forward network (Ponti and Korhonen, 2017), convolutional neural networks (Jin et al., 2020; Kruengkrai et al., 2017) and recurrent neural networks (Yao et al., 2019). Later unsupervised training model such as BERT (Devlin et al., 2018; Sun et al., 2019) and RoBERTa (Becquin, 2020) are also used.

In this work, we base our model on RBMs (Restricted Boltzmann Machine), a deep learner for identifying the cause-effect (arguments) and a CRFs (Conditional Random Fields) Model for identifying the event. The rest of this paper is organized as follows. In Section 2 we present an analysis of causal constructions in Tamil and the data. Section 3 describes the method used for extracting causal relations from News wire text in Tamil. Results and discussions are presented in Section 4. At the end we give the conclusion.

2. Analysis of Cause –Effect in Tamil

The cause-effect relation in Tamil is characterized by the cause, the effect and an optional marker. The marker indicates the presence of a cause-effect relation. The cause is the event that is the reason for the other event called the effect to happen. There is a dependency of one event on the other. One event causes the other event. In other words, an event is a consequence of a preceding event. The cause might be just one of the reasons for the effect to happen in real-world, but what matters in the context of cause-effect relations is the way it is expressed in text. The text may express more than one event as the reasons for the effect to happen, which is a case of multiple causes.

WILDRE-7 Workshop, pages 47–53

2.1 Types of Cause-Effect Relations.

The cause-effect relation in Tamil is classified broadly into explicit cause-effect relations and implicit cause-effect relations. An explicit cause-effect relation is an expression which contains a cause-effect marker explicitly. Certain morphological or syntactic elements bring out the causal meaning. The cause-effect marker denotes the presence of a cause-effect relation. An implicit cause-effect relation is inferred from the context and the world knowledge i.e., there is no explicit cause-effect marker to denote the presence of a cause-effect relation. Ex1 shows an explicit cause-effect relation and Ex.2 shows the same cause-effect relation as in Ex.1, but not connected by an explicit cause-effect marker. Based on the semantics of the context, the reader infers a cause-effect relation. Cause is marked as "C" and effect as "E".

Ex1. [kaaRRu aTi-tt-ataal]C [tuNikaL paRa-nt-ana]E
Wind blow-Pst-Cause clothes fly-Pst-3pn
'Because the wind blew,the clothes flew.'

Ex2. [kaaRRu aTi-tt-atu]C
. [tuNikaL paRantana]E.
Wind blow-Pst-3sn
clothes fly-Pst-3pn
' The wind blew. The clothes flew.'

In the corpus, it was observed that the cause and effect are not always as simple as shown in the examples.

2.2 Text Span of Cause/Effect

The span of text denoting cause or effect does not always coincide with clause boundaries and sentence boundaries. The identification of the text span of the cause and the effect is not very straightforward. The following examples illustrate the point.

Ex3. [atai naan kaNTataal]C ["atellaam nii een paarkkiRaay(finite verb)" enRaaL]E.

'[As I saw that]C, ["Why are you seeing those?" said she']E.

It can be noted that the first finite verb following the causal marker is not necessarily the end of the effect because of the verb occurring within the quotes in direct speech. The text span of cause or effect can stop at the boundary of the first verb in reported speech as well (Ex4).

Ex4.[appaTip paTippataal]C [ivvaLavu aRivu vaLarntiruntat]Eai uNarnteen.

'I realized that [by studying so]C, [my knowledge improved so much]E.

In Ex.4, the effect does not extend up to the end of the sentence. In addition, it can be noted that the end of the text span does not coincide with the end of a token.

2.3 Interdependency of Cause-Effect Relations.

Sometimes cause-effect relations form a chain with the effect of the first relation being the cause of the second. Two cause-effect relations occurring in close proximity can be interdependent.

Ex5. [vaNTikkaararkku ippootu varuvaay kuRaintupoo^nataal]C [[avarkaL kutiraikaLai na^nRaaka vaittiruppatillai]E]C. aakaiyaal [ippootuLLa kutiraikaLum mu^npool paarppataRku azakaaka illai]E.

'[Because the cart-owners' incomes have reduced these days]C, [[they do not care for the horses well] E]C. So, [the horses these days don't look as beautiful as those before] E.'

2.4 Anaphors

Most often, though the cause and effect are found in close proximity to the marker, complete sense cannot be made with this information alone due to the presence of anaphors. Thus anaphors have to be resolved for complete comprehension of the cause-effect relation. In Ex.6, the pronominal anaphors, nii and avan should be resolved to completely understand the two events.

Ex6. [nii anpaaka pazakiyataal]C [avan appaTi eNNiviTTaan]E.

'[Because you interacted lovingly]C, [he thought so]E.'

The above issues are some of the major ones which have to be resolved for identification of the cause and effect of a cause-effect relation. From the linguistic analysis we have arrived at the following

1. A cause-effect relation consists of the cause, the effect and an optional marker and can have multiple causes and/or multiple effects.

2. The cause-effect relation can be classified as explicit and implicit cause-effect relations based on the presence or absence of a marker.

3. In an implicit cause-effect relation, the subordinate clause has a non-finite verb in the infinitive form and Explicit cause-effect relations is marked by a grammatical marker or a lexical marker.

4. Explicit cause-effect markers can be intra-sentential or inter-sentential. Intra-sentential markers can be inter-clausal or intra-clausal. Also Intra-sentential markers are grammatical markers "Grammatical markers" get inflected with a noun or a verb.

5. The grammatical marker for cause-effect that inflect with a noun is -aal. This is a polysemous marker denoting instrumentality and cause-effect among others. This ambiguity in sense is resolved by the verb phrase of the clause in which the marker occurs.

6. The grammatical markers for cause-effect that inflect with a non-finite verb are -ataal, -ata^naal, -ati^naal, -amaiyaal, -aamaiyaal. They denote the cause in the subordinate clause and the effect in the main clause.

7. There are Inter-sentential discourse connectives like ata^naal, ita^naal, aa^napaTiyaal, aakaiyaal, aakaiyi^naal, aatalaal, aakavee, e^navee are lexical markers denoting cause-effect.

8. There are other lexical markers such as kaaraNam, kaaraNamaaka and kaaraNattaal and they occur in complex patterns.

9. Certain verbs inherently denote cause.

2.5 Benchmark Datasets

As we all know that data is the foundation of experiment. There is a number of datasets available for English which are used for evaluating cause-effect models. The

SemEval-2007 task 4, it is part of SemEval (Semantic Evaluation), the 4th edition of the semantic evaluation event (Girjuet.al 2007). This task provides a dataset for classifying semantic relations between two nominals. Within the set of seven relations, the organizers split the Cause–Effect examples into 140 training with 52.0% positive data, and 80 test with 51.0% positive data. SemEval-2010 task 8, unlike its predecessor, SemEval-2007 Task 4, which has an independent binary-labelled dataset for each kind of relation, this is a multi-classification task in which relation label for each sample is one of nine kinds of relations (Hendrickx I 2010). PDTB 2.0, the second release of the penn discourse treebank (PDTB) dataset from Prasad et al. (Prasad R 2007) is the largest annotated corpus of discourse relations. It includes 72,135 non-causal and 9190 causal examples from 2312 Wall Street Journal (WSJ) articles. TACRED, similar to SemEval, the Text Analysis Conference (TAC) is a series of evaluation workshops about NLP research. The TAC Relation Extraction Dataset (TACRED) contains 106,264 newswire and online text that have been collected from the TAC KBP challenge.1 during the year from 2009 to 2014 (Zhang Y 2017). BioInfer (Pyysalo et al.2007) introduce an annotated corpus, BioInfer (Bio Information Extraction Resource), which contains 1100 sentences with the relations of genes, proteins, and RNA from biomedical publications. There are 2662 relations in the 1100 sentences, of these 1461 (54.9%) are causal-effect. ADE, the corresponding ADE task aims to extract two entities (drugs and diseases) and relations about drugs with their adverse effects (ADEs) (Hidey C, and McKeown K 2016).

2.6 Tamil Data

There are no standard annotated dataset for cause-effect for Tamil and for any Indian languages. The data we have used is annotated in house from different genres such as novels and new wires. The details of causal markers and their distribution in the corpus is given below (Table -2). In this work the data is collected through crawling the content from 5 major online Tamil News portals. The data is collected over a period of time, by performing daily crawling. The online News portals used for crawling are listed below:

1. Dinamani – https://www.dinamani.com/
2. Dinathanthi- ttps://www.dailythanthi.com/
3. Dinamalar – https://www.dinamalar.com/
4. The Hindu (Tamil)– https://www.hindutamil.in/
5. Maalaimalar - https://www.maalaimalar.com/

There are a total of 2000 documents. Each document is a News article. The average size of a News article is 25 Sentences. Along with this we have also taken data from a few Tamil story and travelogue blogs and Novels. In Table 1 the data statistics is given. The column #sentences shows the total number of sentences. The second column #Relations, shows the number of causal relations. And Table 2 describes causal markers distribution in the corpus (data statistics based on different causal markers). In this table 2 the second column gives number of times the causal marker has occurred in the corpus and third column gives the number of instances where a cause-effect relation has occurred.

Table 1. **Overall Corpus Statistics**

SNo	Corpus Type	#Sentences	#Relations
1	News Corpus	50300	3590
2	Web blogs	488	45
3	Tamil Novels (Akalvilakku, Civakamiyin Sapatham, Kurinchi Malar)	31741	1345
	Total	**83529**	**4980**

We have annotated the data manually using trained linguists. The cause is marked by "C" and effect by "E". We calculated the inter-annotators agreement using Kappa score and the score was 96%.

Table 2. Causal Markers Distribution in the corpus

SNo	Causal Marker	Total no. of occurrences	No. of Cause-Effect relations
1	*-ataal -atanaal, -itanaal, -paTiyaal, -amaiyaal, -aamaiyaal*	720	660
2	*atanaal, itanaal, aakaiyaal, aanapaTiyaal, aata laal, aakavee, enavee*	1470	1450
3	*kaaraNattaal*	230	210
4	*kaaraNamaaka*	230	210
5	*kaaraNam*	720	490
6	*-aal*	6360	1010
7	*eenenil, eenenRaal*	980	950
	Total	**10710**	**4980**

3. Our Method

In this work we have followed the two step approach,

Step 1: Event Identification using Conditional Random Fields (CRFs), a machine learning algorithm.

Step 2: The Cause-Effects (Causal relations) related to the event are extracted using Restricted Boltzmann Machine (RBM), an unsupervised deep learning algorithm.

Before doing the Event Identification and Cause-Effect identification the documents are pre-processed for syntactic and semantic information enrichment.

Syntactic Pre-Processing: The data obtained from crawling online news portals is cleaned and the text content alone is extracted. After the content is extracted and cleaned, the syntactic pre-processing of the data is performed. We use in house developed POS Tagger (Arulmozhi & Sobha., 2006), and Chunker (Pattabhi et al., 2007) for pre-processing. The text that is split into sentences and then tokenized is send to POS tagger for tagging the POS and the POS tagged data is given to the chynker for chunking. The performance of the POS tagger is 93.26% accuracy and for chunking, the accuracy is 92.73%.

Semantic pre-processing: The semantic pre-processing of the data includes named entity (NE) tagging and anaphora resolution(AR). It is observed that any event there is an involvement of a human/non-human entity, location and time. Thus the entity identification is important.

3.1 Event Identification

The event extraction is performed using Conditional Random Fields (CRFs). The challenge in developing an event extraction system using ML techniques lies in designating the striking features and designing of feature template. A window size of 5 is used in this work. We describe in detail the features used in developing the event identification system.

Lexical features and Syntactic features: Word, Parts of Speech (PoS) and chunk are used. PoS help in disambiguating the sense of the word in a sentence. PoS is an important feature for extracting the events as most of the arguments of an event are proper noun and event trigger belongs to noun and verb category. Hence PoS is a key feature for event extraction task. Most of the event trigger and arguments are descriptive i.e., they occur as a phrase. Hence chunk tag will help in argument and event trigger extraction.

Named Entities (NEs): Named Entity Recognition (NER) is the task of extraction of NEs such as place names, organization names, person names, facilities names etc. from a given text document.

The combination of all the above described features is used to develop the feature template for training the CRFs for Event identification and the language model is obtained.

3.2 Cause-Effect Identification

For each event identified, there will effects of the event and causes of the event. For example for an "earthquake event", the causes could be tectonic plate shifts and effects are the severe damages to the people, animals and other properties, facilities etc. For automatic identification of causal relations Restricted Boltzmann Machine (RBM), an unsupervised deep learning algorithm is used. RBM is a type of Boltzmann Machines (BMs), to make them powerful enough to represent complicated distributions which go from the limited parametric setting to a non-parametric one. We consider that some of the variables are never observed (they are called hidden). By having more hidden variables (also called hidden units), it can increase the modelling capacity of the Boltzmann Machine (BM). Restricted Boltzmann Machines (RBMs) further restrict BMs to those without visible-visible and hidden-hidden connections. Unlike other unsupervised learning algorithms such as clustering, RBMs discover a rich representation of the input. RBMs are two-layer neural nets. The first layer of the RBM is called the visible, or input, layer, and the second is the hidden layer.

These data in the visible layer (or input layer) is converted to vectors of n-dimension and passed to the hidden layer of the RBM. The word vectors are the vector representations. These are obtained from the word2vec. These are also called as word embedding. Word embedding, in computational linguistics, referred as distributional semantic model, since the underlying semantic theory is called distributional semantics. The real valued n-dimensional vector for each level is formed using the word2vec algorithm. Word2vec creates or extracts features without human intervention and it includes the context of individual words/units provided in the projection layer. Word2vec is a computationally-efficient predictive model for learning word embedding's from text. The context comes in the form of multiword windows. Given enough data, usage and context, Word2vec can make highly accurate word associations. Word2vec expects a string of sentences as its input. Each sentence – that is, each array of words – is vectored and compared to other vectored lists of words in an n-dimensional vector space. Related words and/or groups of words appear next to each other in that space. The output of the Word2vec neural net is a vocabulary with a vector attached to it, which can be fed into the next layer of the deep-learning net for classification. We make use of the DL4J Word2vec API for this purpose [Mikolov 2013].

We have obtained optimal hyper parameters for good performance by performing several trials. The main hyper parameters which we need to tune include choice of activation function, number of hidden units, learning rate, dropout value, and dimensionality of input units. We used 20% of training data for tuning these parameters. The optimal parameters include: 200 hidden units, rectilinear activation function, 200 batch size, 0.025 learning rate, 0.5 dropout and 25 training iterations. We obtained best development set accuracy at 80 dimensional words. The output layer uses Softmax function for probabilistic multi-class classification. The Softmax function classifies into two classes: A Causal relationship (Cause/Effect) or not a causal relationship. We train the RBM and using the language model obtained during the training of RBMs on a given new text document.

4. Results and Discussion

This section describes the performance of our system in terms of Precision, Recall and F-score. Precision is the number of Events/Causal relations correctly identified by the system from the total number of Events/Causal relations identified by the system. available in the gold standard. Recall is the number of Events/Causal relations correctly detected by the system to the total number of Events/Causal relations available in the corpus (gold standard). F-score is the harmonic mean of precision and recall.

$$\text{Precision} = TP / (TP + FP)$$
$$\text{Recall} = TP / (TP + FN)$$
$$\text{F score} = (2 \times \text{Recall} \times \text{Precision} / (\text{Recall} + \text{Precision}))$$

where, TP means true positives, FN means false negatives and FP means false positives.

In this work for the evaluation, only the single causal relations are considered and not the embedded ones. A causal relation that is inside or embedded inside another causal relation is not considered as separate a distinct causal relation.

A 10-fold cross validation experiment is performed on the data, by splitting the data into 10 equal parts. A set of 9 equal parts is concatenated to form training partition and 1 part is used for testing. Table 3 describes the results obtained for 10-fold cross validation experiment. We have obtained an average precision of 84.35% and average recall of 81.04%.

Table 3. Causal Relation – 10 Fold Experiment results

n-Fold Number	Total C-E Relations in the test set (Gold tagged)	Total C-E Relations Identified by the system	Total C-ERelations Correctly identified by the system	Precision %	Recall (%)
1	460	451	367	81.37	79.75
2	410	401	325	81.05	79.18

3	424	407	335	82.31	79.03
4	454	437	369	84.42	81.23
5	446	425	377	88.70	84.55
6	498	467	399	85.44	80.18
7	464	459	389	84.75	83.78
8	476	474	387	81.64	81.32
9	497	463	403	87.04	81.07
10	418	387	336	86.82	80.33
			Average	**84.35**	**81.04**

5. Conclusion

We have described in detail the cause and effect in Tamil with linguistic analysis, how automatically the cause and effect can be identified using a 2 step process where we have used CRFs to identify the events and RBM for identify the cause and effect of the event. The system works on News paper articles and it is a real time application. The system works with 84.35% precision and 81.04% recall.

6. Acknowledgments

We acknowledge Bhashini, NLTM project under MeitY on Discource Integrated Dravidian to Dravidian Machine Translation (DL-DiscoMT) for the funding to do this research.

7. Bibliographical References

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In Proc. of the 14th conference on Computational linguistics, pages 539–545.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pages 1–8.

Daniela Garcia et al. 1997. Coatis, an nlp system to locate expressions of actions connected by causality links. In International Conference on Knowledge Engineering and Knowledge Management, pages 347– 352. Springer.

Christopher SG Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In Proceedings of the 38th annual meeting of the association for computational linguistics, pages 336–343.

Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. 2012. Learning causality for news events prediction. In Proceedings of the 21st international conference on World Wide Web, pages 909–918

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2009. Classification of semantic relations between nominals. Language Resources and Evaluation, 43(2):105–121.

Ning Kang, Bharat Singh, Chinh Bui, Zubair Afzal, Erik M van Mulligen, and Jan A Kors. 2014. Knowledge-based extraction of adverse drug events from biomedical text. BMC bioinformatics, 15(1):1– 8.

Quoc-Chinh Bui, Breanndán Ó Nualláin, Charles A Boucher, and Peter Sloot. 2010. Extracting causal relations on hiv drug resistance from literature. BMC bioinformatics, 11(1):1–11.

Quang Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 294– 303.

Edoardo Ponti and Anna-Leena Korhonen. 2017. Eventrelated features in feedforward neural networks contribute to identifying implicit causal relations in discourse.

Xianxian Jin, Xinzhi Wang, Xiangfeng Luo, Subin Huang, and Shengwei Gu. 2020. Inter-sentence and implicit causality extraction from chinese corpus. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 739–751. Springer.

Canasai Kruengkrai, Kentaro Torisawa, Chikara Hashimoto, Julien Kloetzer, Jong-Hoon Oh, and Masahiro Tanaka. 2017. Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 31.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 7370–7377.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language

Cong Sun, Zhihao Yang, Ling Luo, Lei Wang, Yin Zhang, Hongfei Lin, and Jian Wang. 2019. A deep learning approach with deep

contextualized word representations for chemical–protein interaction extraction from biomedical literature. IEEE Access, 7:151034–151046.

Guillaume Becquin. 2020. Gbe at fincausal 2020, task 2: span-based causality extraction for financial documents. In Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 40–44.

Sindhuja Gopalan, Sobha Lalitha Devi 2017 Cause and Effect Extraction from Biomedical Corpus Comp. y Sist. vol.21 no.4 Ciudad de México oct./dic. https://doi.org/10.13053/cys-21-4-2854

Menaka. S, Rao, P.R.K., Lalitha Devi, S. (2011). Automatic Identification of Cause-Effect Relations in Tamil Using CRFs. In: Gelbukh, A.F. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2011. Lecture Notes in Computer Science, vol 6608. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19400-9_25

Roxana Girju, Nakov P, Nastase V, Szpakowicz S, Turney P, Yuret D (2007) Semeval-2007 task 04: classification of semantic relations between nominals. In: Proceedings of the 4th international workshop on semantic evaluations. Association for Computational Linguistics, USA, SemEval '07, pp 13–18

Hendrickx I, Kim SN, Kozareva Z, Nakov P, Ó Séaghdha D, Padó S, Pennacchiotti M, Romano L,Szpakowicz S (2010) SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, pp 33–38

Prasad R, Miltsakaki E, Dinesh N, Lee A, Joshi A (2007) The penn discourse treebank 2.0 annotation manual. IRCS technical reports series 203 Philadelphia: University of Pennsylvania ScholarlyCommons, p 105

Zhang Y, Zhong V, Chen D, Angeli G, Manning CD (2017) Position-aware attention and supervised data improve slot filling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, pp 35–45. https://doi. org/10.18653/v1/D17-1004 [97]

Pyysalo S, Ginter F, Heimonen J, Björne J, Boberg J, Järvinen J, Salakoski T (2007) Bioinfer: a corpus for information extraction in the biomedical domain. BMC Bioinform 8:50. https://doi.org/10.1186/1471-2105-8-50

HideyC, McKeown K (2016) Identifying causal relations using parallel Wikipedia articles. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1:longpapers).Association for computational linguistics, Berlin, Germany, pp 1424–1433. https://doi.org/10.18653/v1/P16-1135

Creating Corpus of Low Resource Indian Languages for Natural Language Processing: Challenges and Opportunities

Pratibha Dongare
The English and Foreign Languages University, Hyderabad, India.
pratibhaphdlandp22@efluniversity.ac.in

Abstract

Addressing tasks in Natural Language Processing requires access to sufficient and high-quality data. However, working with languages that have limited resources poses a significant challenge due to the absence of established methodologies, frameworks, and collaborative efforts. This paper intends to briefly outline the challenges associated with standardization in data creation, focusing on Indian languages, which are often categorized as low resource languages. Additionally, potential solutions and the importance of standardized procedures for low-resource language data are proposed. Furthermore, the critical role of standardized protocols in corpus creation and their impact on research is highlighted. Lastly, this paper concludes by defining what constitutes a corpus.

Keywords: Low resource languages, Corpus, Indian Languages

1. Introduction

Natural Language Processing (NLP) has witnessed unprecedented growth and advancements in machine learning, artificial intelligence and other allied fields. However, while NLP models have flourished in well-resourced languages, the landscape becomes markedly challenging when operating within low-resource language domains. Data plays a crucial role in any NLP task. The quality and quantity of the data has a huge impact on the performance of a system. The type of corpus may vary according to the tasks. For instance, spoken, textual, conversational, lexical, learner, and other types of corpora can be used while working on TTS, ASR, information extraction tasks, discourse corpus or conversational corpus can be used in creating chatbots or training LLMs, parallel corpus can be used in Machine translation.

India is a diverse country with many languages, but it lacks the necessary resources to adequately support even the most widely spoken Indian languages. When considering Asia as a whole, which is linguistically dense, similar challenges arise in representing these languages computationally. The absence of fundamental NLP tools for these languages has significant social implications (Singh, 2008).

Low-resource languages, commonly characterized by limited availability of linguistic

data and tools, pose unique obstacles in developing effective NLP solutions. Low-resource languages are also known as less privileged languages (Singh, 2008), less advanced languages (Dash and Ramamoorthy, 2019), under-resourced, and resource-poor languages. Low resource languages can be understood as less studied, resource scarce, less computerized, less privileged, less commonly taught or low density among other denominations (as cited in Maguersse et al., 2020; Singh, 2008; Cieri et al., 2016; Tsvetkov, 2017).

When examining the definition of the concept, it becomes evident why Indian languages are classified as low-resource languages. Atkins et al. (1992) outlined several challenges experienced by languages with varying levels of development, from less advanced to more advanced languages. When exploring the lack of resources in languages, various factors come into play:

- **Digital presence:** Digital representation encompasses the online presence of a language, including its information in various domains, subjects, and diverse forms of data. It is crucial to gauge the extent of this data to address the ongoing debate about what constitutes an adequate amount of information.
- **User friendliness:** It is crucial to achieve a harmony between usability and linguistic representation. According to the findings of the KPMG- Google (2017) survey, it is projected that by 2021, 8 out of 10 Indian language users will access the Internet in regional Indian languages. This development significantly influences the accessibility and user-friendliness of these

WILDRE-7 Workshop, pages 54–58

languages. Therefore, it is imperative to invest in the development of language resources and NLP tools for low-resource languages to ensure equal opportunities and benefits for all linguistic communities.

- **Language Processing and Tools:** Enabling the development of language processing tools becomes feasible with increased availability of data. The foundation of NLP research relies on the presence of corpora; structured datasets (written, spoken, multimodal, etc.) carefully selected for training and evaluating language models. Corpora form the basis for various NLP applications such as machine translation, sentiment analysis, and information extraction. However, the creation and standardization of corpora poses complex challenges, particularly in languages with limited resources.

Corpus-based studies are incorporating new insights to investigate the cognitive areas of the human mind to understand the mysteries operating behind the cognitive process like receiving, processing, comprehending, and sharing linguistic signals (Winograd, 1983). Corpus can be used in wide applications. For instance, domains of social sciences, machine learning, sentiment analysis, dictionary compilation, grammar writing, wordnet design, word-sense disambiguation, translation, documentation, and other areas of linguistics like diachronic lexical semantics, pragmatic analysis of texts, sociolinguistic studies, and discourse analysis (Dash and Arulmozi, 2018; Leech and Fligestone, 1992).

Compared to other countries, India lags far behind not only in corpus generation but also in corpus-based linguistic studies and application Dash and Ramamoorthy (2019). The next section briefly outlines the challenges faced while working on resource-poor languages.

2. Challenges

Creating and compiling the corpus presents numerous challenges, some of which are briefly outlined in this section.

1. **Data scarcity:** As mentioned in the previous section, a significant challenge for Indian languages is the limited availability of resources, including corpora and tools. Despite an increase in internet and technology users, there has been no corresponding increase in resources for regional languages. Therefore, the development of resources for these languages presents a significant challenge. The accessibility of various domains and topics is also extremely important. The lack of diverse and representative data in the corpora for regional languages is another challenge. For example, there are several freely available datasets for download and use, such as those developed AI4 Bharat[1], datasets available on TDIL[2] and LDCIL[3] portals. However, these datasets primarily emphasize the scheduled languages and cover a limited range of domains like news articles. Furthermore, the lack of standardization and documentation poses difficulties in corpus compilation.
2. **Quantity of data:** Determining the appropriate amount of data is an important yet debatable question when it comes to building a corpus. The emergence of GPT models has recently generated considerable interest in large datasets, but gathering extensive data for languages with limited digital presence remains a challenge. Dash and Ramamoorthy (2019) have emphasized that the distribution of written and published texts is uneven, posing a challenge for corpus compilation. While Sinclair (19991) stated that containing around 1 million words may be sufficient for specific linguistic studies and research, Dash and Ramamoorthy argue that at least 10 million words are necessary for language description purposes.
3. **Linguistic and non-linguistic challenges:** Numerous Indian languages exhibit diverse varieties and dialects, with a significant number of speakers. Hence, it is imperative to address the need to support these various dialects and linguistic features. Furthermore, consideration must also be given to the shared linguistic attributes among these languages, their scripts, and variations. Additionally, certain tribal languages lack scripts altogether, necessitating representation using alternative available scripts while some languages use multiple scripts. For instance, Korku and Munda languages (languages belong to Austro Asiatic family of languages) have no regular scripts whereas Santali (one of the scheduled languages of India) uses five scripts:

[1] https://ai4bharat.iitm.ac.in/resources/datasets/

[2] https://tdildc.in/index.php?option=com_download&task=fsearch&Itemid=547&lang=en

[3] https://data.ldcil.org/index.php?route=common/home

Devanagari, Bengali, Odia, Alchiki and Roman scripts (census of. India, 2022).

4. **Code mixing:** The growing trend of code mixing, in which individuals alternate between two or more languages within a single conversation or sentence, presents challenges in compiling corpora and analyzing language. The use of script mixing (romanization) poses significant difficulties when dealing with data from social media platforms. Transliteration between roman and regional scripts, as well as glossing and annotation, becomes essential to account for code mixing during corpus development. For instance, the sentence, *'मी try करेन' (Mi try karen) (I will try)* uses Marathi and English with both Devanagari and roman script, requiring transliteration and annotation to accurately represent code mixing.

5. **Computational assistance:** Data must be represented, stored, and managed using computational devices. Advances in hardware and software technologies have facilitated data optimization. However, limited knowledge of these technologies and the affordability of such devices pose challenges. It is important for a wide range of researchers to have access to the latest hardware systems, updated software versions, and operating systems. For example, the availability of support for the OCR technique is also limited, which affects the digitization of many regional language texts. For example, old Marathi texts and manuscripts are written using a script called moḍi. Although this script is not commonly used today, it still holds significance in facilitating computational assistance for preserving languages and conducting diachronic research.

6. **Standardization challenges:** Supporting LR languages presents significant challenges such as transcription, transliteration, glossing, and encoding. Using a widely accepted standard such as Unicode facilitates consistent data representation, ensuring accurate display and processing across different software and hardware platforms. The absence of such standardization complicates the creation of a reliable corpus, which requires additional conversion efforts to integrate data from diverse sources into the Unicode-based corpus. Indic languages typically utilize 8-bit fonts for encoding. However, despite the existence of a standard 8-bit code table and layout for Devanagari in ISCII, varying keyboard layouts and non-standard character sets employed by font designers contribute to difficulties in standardization when gathering data from multiple sources (McEnery et al., 2000).

7. **Data revision and updates:** The data must undergo regular revisions and updates to ensure the relevance and accuracy of the information. This is particularly important for LR languages, as acquiring the initial data poses a significant challenge. Consequently, maintaining data quality and implementing updates presents an even greater challenge. For instance, machine learning models heavily rely on training data, regular updates are necessary to adapt to evolving language patterns and improve performance.

8. **Ethical considerations:** When conducting research on lesser-represented languages, particularly involving speech corpus, it is crucial to prioritize ethical considerations. This applies not only to multimodal data, but also when working with smaller language communities and non-mobile populations. It is essential to take steps to ethically collect and document high-quality data in these cases.

3. Potential Solutions

Working with LR languages presents various challenges, and the following section emphasizes the importance of standardization while offering potential solutions to these obstacles.

- In order to create uniformity in generation, compilation, and maintenance of the corpus, we need a standardized procedure or common guidelines. For instance, in Baker et al. (2003) mentioned that in their study, ISCII was an attempt to standardize 8-bit encodings for Indian writing systems, but the paper notes that this standard is largely ignored by developers of TTF fonts for Indic scripts and so is mostly absent from the web. This leads to a significant challenge in corpus creation, as many different incompatible glyph encodings exist for Indic fonts compared to a standardized approach, like the hexadecimal code 42 always representing "B" in English fonts. ASCII is a character encoding standard for electronic communication that represents text in computers, and UTF-8 is a variable width character encoding that can represent all

characters in the Unicode character set. Both play crucial roles in text processing and data exchange, with UTF-8 being particularly important as a way to encode Unicode characters efficiently while preserving backward compatibility with ASCII. Baker et al (2003 a) conducted research on the EMILLE corpus. They emphasized that transforming numerous 8-bit based texts into a uniform format such as Unicode was challenging and time-consuming, mainly because of the absence of consistent 8-bit font encoding standards across various creators of electronic texts in the respective languages. This proved to be a substantial technical obstacle in compiling the corpora. In the proposed solution, McEnery and colleagues (2000) used a 16-bit universal character set.

- The standardized process used in creating and managing the corpus, along with encoding, will assist linguists and annotators by providing a clear framework for collaboration. This will facilitate the development of consistent guidelines for data annotation, preprocessing, and analysis to ensure high-quality results.
- Working with a standardized approach for low-resource languages is crucial as it would not only support the computational advancement of these languages but also enable more widespread contributions. Additionally, this approach would make it possible to have a comparative analysis of the data, leading to valuable insights and progress in linguistic research.
- **Interoperability:** Interoperability can be improved through the establishment of standard and uniform procedures. This would lead to better data transfer and usage, benefiting researchers worldwide. Moreover, this improvement in interoperability would ensure greater convenience regardless of costly hardware or software upgrades. Furthermore, adopting a standardized approach in text processing and data exchange would promote accessibility and inclusivity.
- **Quality of data:** When creating a corpus, it is crucial to take into account different linguistic and statistical factors like the size of the data, its manner, intended users or tool usage (in relation to task- specific and domain-specific tools), multilingual and monolingual data, preprocessing and cleaning procedures, as well as data storage and management. Ethical considerations are necessary to ensure the authenticity of the data by obtaining prior consent from participants or informants. It is important to also consider potential biases in the collection process that might affect the overall quality of the corpus.
- **Collaborative efforts:** Collaboration and contribution from researchers with diverse expertise in linguistic, statistical, and computational fields are essential for the development and advancement of LR languages. Through their combined efforts, we can achieve more accurate, dynamic, and impactful results. Advancements in the field will be achieved through community efforts.

4. Conclusion

The paper aimed to briefly outline the practical obstacles encountered when working with Indian language corpora. The compatibility, accessibility, and interoperability of the data can be improved using the standard practices and efforts. The potential solutions could be improved. It is necessary to consider multidimensional and multilingual corpus development, which can have applications in various related fields such as language description, comparative analysis, documentation, tool development, and more. A corpus is not simply a collection of data; rather, it is a curated and processed collection of information tailored for specific research purposes because, while gathering data may not be challenging, transforming it into a corpus is.

5. Bibliographical References

Atkins, S., Clear, J., & Ostler, N. (1992). Corpus design criteria. *Literary and linguistic computing*.

Baker, P., Hardie, A., McEnery, T., & Jayaram, B. D. (2003a). Corpus data for South Asian language processing. In *Proceedings of the 10th Annual Workshop for South Asian Language Processing, EACL*.

Baker, P., Hardie, A., McEnery, T., & Jayaram, B. D. (2003a). Constructing corpora of South Asian languages. https://ucrel.lancs.ac.uk/publications/cL2003/papers/baker.pdf

Census of India. (2022). Census of India 2011—*Language Atlas- India.* Office of the Registrar General & Census Commissioner, India. https://censusindia.gov.in/nada/index.php/catalog/42561

Dash, N. S., & Arulmozi, S. (2018). *History, features, and typology of language corpora*. Springer Singapore.

Dash, N. S., & Ramamoorthy, L. (2019). *Utility and application of language corpora.* Singapore: Springer.

KPMG-Google. (2017). *Indian languages defining India's internet.* https://assets.kpmg/content/dam/kpmg/in/pdf/2017/04/Indian-languages-Defining- Indias-Internet.pdf

Leech, G. and S. Fligestone. (1992). *Computers and corpus analysis*. Oxford: Blackwell publishers.

Magueresse, A., Carles, V., & Heetderks, E. (2020). Low-resource languages: a review of past work and future challenges. *arxiv preprint arxiv:2006.07264*.

McEnery, A., Baker, P., Gaizauskas, R., & Cunningham, H. (2000). EMILLE: Building a corpus of South Asian languages. In *Proceedings of the International Conference on Machine Translation and Multilingual Applications in the new Millennium: MT 2000*.

Sinclair, J. (1991). *Corpus, concordance, collocation.* Oxford University Press.

Singh, A. K. (2008). Natural Language Processing for Less Privileged Languages: Where do we come from? Where are we going?. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

Tsvetkov, Y. (2017). Opportunities and challenges in working with low-resource languages.

http://www.cs cmu.edu/~ytsvetko/jsalt-part1.pdf.

Winograd, T. (1983). *Language as a cognitive process.* Vol. I. Mass: Addison-Wesley publication.

FZZG at WILDRE-7: Fine-tuning Pre-trained Models for Code-mixed, Less-resourced Sentiment Analysis

Gaurish Thakkar, Marko Tadić, Nives Mikelić Preradović
Faculty of Humanities and Social Sciences, University of Zagreb
{gthakkar, marko.tadic, nmikelic}@ffzg.unizg.hr

Abstract

This paper describes our system used for a shared task on code-mixed, less-resourced sentiment analysis for Indo-Aryan languages. We are using the large language models (LLMs) since they have demonstrated excellent performance on classification tasks. In our participation in all tracks, we use *unsloth/mistral-7b-bnb-4bit* LLM for the task of code-mixed sentiment analysis. For track 1, we used a simple fine-tuning strategy on PLMs by combining data from multiple phases. Our trained systems secured first place in four phases out of five. In addition, we present the results achieved using several PLMs for each language.

Keywords: sentiment analysis, code-mixed, LLM, Indo-Aryan

1. Introduction

Expression of sentiment-bearing information is natural to humans. The information expressed can span a spectrum of positive, negative, neutral, and mixed connotations. Sentiment (Turney, 2002) plays a major role in the interaction of people through social media as a tool of expression. Social media has evolved as an effective tool for people to express their views and ideas on a wide range of issues (Alodat et al., 2023; Kapoor et al., 2018). The interaction of social media users around the world has led to numerous phenomena (Nasir Ansari and Khan, 2020). One of them is code-mixing, also called intra-sentential code switching or intra-sentential code alternation and it occurs when speakers use two or more languages below clause level within one social situation (Mónica et al., 2009). For instance, the phrase "**Superhit bahut Achcha**" translates to "*superhit very good*" in English. The phrase is written in Roman characters instead of Devanagari and incorporates terms from both English and Hindi. This example does not necessarily follow standard writing rules (Das and Gambäck, 2014), but it effectively demonstrates its amalgamating nature, it poses a significant problem to process this text as it contains language constructs borrowed from multiple languages. Therefore, it is important to develop systems that can handle these phenomena to better understand sentiment. The code-mixed dataset can be understood by individuals who understand both languages; hence, developing the system for modelling can be challenging.

The WILDRE-7 shared task was organised for language pairs and triplets of less-resourced closely related languages: Magahi-Hindi-English (Rani et al., 2024a), Maithili-Hindi (Rani et al., 2024b), Bangla-English-Hindi (Raihan et al., 2023), and Hindi-English. Each code-mixed comment or sentence in Magahi-Hindi-English and Hindi-English had been annotated with four sentiment labels (positive, negative, neutral or mixed). However, the Bangla-English-Hindi is labelled with only three sentiment labels (positive, negative, or neutral).

Our approach to the code-mixed sentiment classification is to use the entire data in a multilingual training setup to aid transfer-learning between languages. The multilingual training helps low-resourced languages owing to the sharing of features between instances of different languages (Schmidt et al., 2022; Alves et al., 2023; Thakkar et al., 2021). We explore three large language models with fine-tuning setups. We combine all the data from different phases into a single dataset and fine-tune two XLM-RoBERTa-based models (Conneau et al., 2020; Barbieri et al., 2022) and one quantized version of the Mistral-7b model (Jiang et al., 2023).

Our final submission for all the phases used supervised fine-tuning on the "unsloth/mistral-7b-bnb-4bit" model[1]. Our proposed model performed well in the Bangla-English code-mixed and combined code-mixed phases. In other phases, despite achieving the best scores compared to other participants, the performance for the relevant languages in the test set was below 0.50 F1.

2. Related Work

An initial investigation into the code switching phenomenon was conducted by Warschauer et al. (2002). They investigated the use of English and Arabic by a group of youthful professionals in email correspondence. It was discovered that English

[1] https://huggingface.co/unsloth/mistral-7b-bnb-4bit

WILDRE-7 Workshop, pages 59–65

was used more frequently in both formal (business-related) email exchanges and Internet searches.

Chittaranjan et al. (2014) employed word-level language identification in code-mixed texts, in which various characteristics were utilised to identify the language of a given word. Contextual features, capitalization features, special character features, and lexicon features were all implemented by the system. Annotated data is then utilised to train the CRF model. The authors attained results with high precision for the majority of language pairs. The accuracy was compromised when the distribution of languages in the test data differed from that of the training data.

Veríssimo dos Santos Neto et al. (2020) proposed, for the Semeval 2020 submission (shared task 9), a combination of four models predicated on the application of transfer learning and language models. The task required conducting sentiment analysis on code-mixed languages that combine English and Hindi. Ma et al. (2020) presented a novel approach in SemEval-2020 for sentiment analysis problem by utilising weighted loss of several multilingual models, with a specific emphasis on the difficulty of code-mixing phrases. The authors employed XLM models in conjunction with machine translation as a form of data augmentation.

3. System Overview

In this section, we describe the task, the different LLMs used, along with preprocessing steps and training configurations.

3.1. Task description

The task had two different evaluation tracks. Track 1 dealt with the classification of the polarity (positive, negative, neutral or mixed) of the comment in the code-mixed setting for the following phases.

1. Hindi-English
2. Magahi-Hindi-English
3. Bangla-English
4. Combined all the language pairs/triplets (1+2+3)

In Track 2, the task was to use the given unlabeled test data for the code-mixed Maithili language (Maithili-Hindi-English) and leverage any or all of the available training datasets in Track 1 to determine the sentiment of a comment in the target language. The dataset was divided into the train, validation and test sets with a ratio of 70:15:15. However, for the fourth part of Track 1 (combining all the language pairs), we combined the provided training and validation datasets of each code-mixed language to train the model.

3.2. Approach

We experimented with two approaches: supervised fine-tuning (Severyn and Moschitti, 2015) and instruction tuning (Efrat and Levy, 2020). Instruction tuning involves providing the model with a collection of instructions or prompts and subsequently modifying the model's parameters to enhance its performance on the tasks specified by these instructions. One way to do this is through the use of techniques such as reinforcement learning (Bai et al., 2022), in which the model receives rewards for behaviours that result in favourable outcomes, or gradient descent (Chen et al., 2022), in which the model's parameters are continuously modified to minimise a loss function.

The following insights served as the foundation for our instruction tuning strategy. For several benchmark datasets, the models (Touvron et al., 2023; Jiang et al., 2023) that were trained using instruction tuning were at the top of the Open LLM Leaderboard[2]. Given that the training cases in the competition were annotated at the sentence level, we concentrated on representing the problem as a single task classification problem without exploring other sub-tasks such as language identification and classification. Since the non-quantized version of Mistral requires extensive processing capabilities, we used the quantized version that can be effortlessly trained on a single GPU with 24 GB of memory.

3.3. Dataset

The organisers provided a dataset (Rani et al., 2024a) containing Magahi-Hindi-English and Hindi-English, which was collected from various YouTube channels and annotated with the help of native speakers of the language. For Bangla-English code-mixed data set 1, we are using the SentMix-3L dataset (Raihan et al., 2023). Table 1 shows the statistics of the provided dataset. In addition, we used SAIL 2017 (Patra et al., 2018), a Hindi code-mixed shared task dataset. In Table 2, the number of instances from the SAIL 2017 dataset is presented.

3.4. Pretrained language models (PLMs)

3.4.1. XLM-RoBERTa-base

XLM-RoBERTa (Conneau et al., 2020) is pretrained on a vast text and code dataset, which includes BooksCorpus, Wikipedia, and the Pile. This

[2]https://tinyurl.com/3s3zfsu8

Phase	Pos	Neg	Neu	Mix
Ben-Eng	293	247	163	
Hin-Eng	1989	419	77	113
Mag-Hin	615	194	26	30
Total	2806	860	266	143

Table 1: Distribution of the dataset released by the organisers.

Split	Pos	Neg	Neu
train	3190	2312	4577
test	399	290	573

Table 2: Additional dataset used for training - SAIL 2017 (Patra et al., 2018)

pre-training technique combines language modelling with natural language task-specific cues, resulting in increased performance on a wide range of activities. It builds on RoBERTa's (Zhuang et al., 2021) great performance by offering new architectural advancements, such as larger model sizes and additional training data. This leads to improved accuracy and efficiency on many NLP tasks.

3.4.2. cardiffnlp/twitter-roberta-base-sentiment

Twitter-RoBERTa-base-sentiment[3] (Camacho-Collados et al., 2022; Loureiro et al., 2022) is a RoBERTa (Zhuang et al., 2021) model trained on ≈124M tweets from January 2018 to December 2021, and fine-tuned for sentiment analysis with the TweetEval benchmark (Barbieri et al., 2020).

3.4.3. unsloth/mistral-7b-bnb-4bit

The Mixtral-8x7B Large Language Model (LLM) is a pre-trained generative Sparse Mixture of Experts. The unsloth/mistral-7b-bnb-4bit model is quantized model of Mixtral-8x7B that has been saved as a LoRA (Hu et al., 2022) adapter through the Unsloth library[4]. The LoRA weights can be retrained during the fine-tuning phase. The model supports a maximum sequence length of 2048, which works optimally with larger contexts.

3.5. Data preparation

In order to generate the training set, we combine all of the code-mixed training sets. We also merge the validation sets of all the datasets provided as part of the competition to create a single validation set. In addition, we incorporate the SAIL 2017 (Patra et al., 2018) dataset as an additional resource into the training to increase the training data size for training the Hindi-English code-mixed model.

3.5.1. XLM-RoBERTa and Twitter-RoBERTa

No special format is required for fine-tuning the model other than tokenizing the dataset with the respective pre-trained tokenizer.

3.5.2. Mistral-7b model

The fine-tuning of the dataset is performed in the form of Instructions. We followed the Alpaca (Taori et al., 2023) dataset format and converted the dataset into the following format:

```
Instruction: Classify the given
article as either positive or
negative or neutral or mix sentiment.

alpaca_prompt = """Below is an
instruction that describes a
task, paired with an input that
provides further context. Write a
response that appropriately
completes the request.

### Instruction:
{Classify the given article as either
positive or negative or neutral
or mix sentiment}

### Input:
{Ekdam sahi bat bolalahi bhaiya}

### Response:
{positive}"""
```

The sentence "**Ekdam sahi bat bolalahi bhaiya**" (hi-en) can be translated to "*You said the right thing brother*" (en). The expected input to the LLM is a single tuple consisting of a prompt, instruction, input, and response. The prompt was the description of the task, the instruction was set to the classification of the text, the input was defined as the code-mixed text, and the response was the expected sentiment label.

4. Experimental Setup

4.1. Fine-tuning

For fine-tuning XLMR models, we used a learning-rate of $5e^{-5}$ with a batch size of 16 and a maximum sequence length of 512. We trained for a maximum of 16 epochs with early stopping and a patience of 3 on the validation set. We used the weighted cross-entropy loss to handle the class imbalance.

[3] cardiffnlp/twitter-roberta-base-sentiment
[4] https://github.com/unslothai/unsloth

4.2. Instruction tuning

For instruction tuning (Efrat and Levy, 2020; Mishra et al., 2022), we used a batch size of 8 and a gradient accumulation of 2. The learning rate was set to $2e^{-5}$ after a few trials. We used the maximum sequence length of 2048. An early stopping mechanism based on a validation set was used to prevent model overfitting.

5. Results

Table 3 presents the initial experiments conducted with the XLM-RoBERTa models. We found that the XLM-RoBERTa performed better than Twitter-RoBERTa, even though Twitter-RoBERTa is trained with Twitter data. The evaluation scores on the target language validation set when using unsloth/mistral-7b-bnb-4bit were better compared to XLM-RoBERTa models.

Model	Eval-F1
XLM-RoBERTa	0.60
Twitter-RoBERTa	0.54

Table 3: Evaluation F-1 scores.

Tr	Phase	F1	P	R
1	Ben-Eng (all)	0.97	0.97	0.97
1	Hin-Eng (all)	0.43	0.50	0.44
	Hin-Eng (Hi+SAIL)	0.44	0.48	0.43
	Hin-Eng (Hi)	0.54	0.54	0.56
1	Mag-Hin-Eng (all)	0.45	0.44	0.57
1	Combined (all)	0.60	0.64	0.57
2	Mai (Hi+SAIL)	0.49	0.45	0.59

Table 4: Final scores reported by the submission system. The scores are reported using predictions obtained using 'unsloth/mistral-7b-bnb-4bit'. The first column (Tr) denotes the track's task number. 'all': A combined training set from the shared task was used for training.

In Table 4, we present the results for the instruction tuning experiments. The model, trained using a combined training dataset, demonstrated strong performance on the test set for Bangla-English, Magahi-Hindi-English, and in combination code-mixed setting. The model achieved higher scores in the Hindi-English test case when exclusively trained on Hindi-English cases. We also attempted alternative combinations, but none of them yielded superior results compared to only using the data instances given as part of the shared task. The findings align with prior research (Thakkar et al., 2021, 2023) indicating that including data from comparable languages with a larger number of training instances improves performance in the case of lower-resourced languages. However, when data instances from lower-resourced languages are combined with higher-resourced languages, there is a decrease in performance for the latter. The combination of the SAIL dataset with Hindi-English training examples was found to be effective combination for training the model to be tested on Hindi-English and Maithili test set.

6. Conclusion

This paper describes the proposed model used for a shared task on code-mixed, less-resourced sentiment analysis for Indo-Aryan languages. We experimented with PLM-based XLM-Roberta and a customised version of Mistral-7b to model the task of code-mixed sentiment. Our analysis shows that code-mixed, less-resourced sentiment analysis for Indo-Aryan languages is a difficult task for the PLMs, and there is scope for further improvements that we will take up in future works. For future work, we would like to use other available code-mixed datasets to improve the performance of sentiment analysis systems in code-mixed settings.

7. Acknowledgements

This work was partially funded by the European Union under the grant agreements No. LC-01641480 – 101018166 (ELE) and No. LC-01884166 – 101075356 (ELE 2). This work was partially funded from the European Union's Horizon Europe Research and Innovation Programme under Grant Agreement No 101070631 and from the UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee (Grant No 10039436).

8. Bibliographical References

Abdelsalam M. Alodat, Lamis F. Al-Qora'n, and Muwafaq Abu Hamoud. 2023. Social Media Platforms and Political Participation: A Study of Jordanian Youth Engagement. *Social Sciences*, 12(7):1–19.

Diego Alves, Božo Bekavac, Daniel Zeman, and Marko Tadić. 2023. Corpus-based syntactic typological methods for dependency parsing improvement. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 76–88, Dubrovnik, Croatia. Association for Computational Linguistics.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn

Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

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, Marseille, France. European Language Resources Association.

Jose Camacho-Collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa Anke, Fangyu Liu, Eugenio Martínez Cámara, et al. 2022. TweetNLP: Cutting-edge natural language processing for social media. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–49, Abu Dhabi, UAE. Association for Computational Linguistics.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identification using CRF: Code-switching shared task report of MSR India system. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 73–79, Doha, Qatar. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387, Goa, India. NLP Association of India.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *CoRR*, abs/2010.11982.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Kawal Kapoor, Kuttimani Tamilmani, Nripendra Rana, Pushp Patil, Yogesh Dwivedi, and Sridhar Nerur. 2018. Advances in social media research: Past, present and future. *Information Systems Frontiers*, 20:531–558.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Yili Ma, Liang Zhao, and Jie Hao. 2020. XLP at SemEval-2020 task 9: Cross-lingual models with focal loss for sentiment analysis of code-mixing language. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 975–980, Barcelona (online). International Committee for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Stella Mónica, Mónica Cárdenas-Claros, and Neny Isharyanti. 2009. Code switching and code mixing in internet chating: betwen "yes", "ya", and "si" a case study. *The jaltcall Journal*, Vol 5:67–78.

Jamal Nasir Ansari and Nawab Khan. 2020. Exploring the role of social media in collaborative learning the new domain of learning. *Smart Learning Environments*, 7(1):9.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task @icon-2017.

Md Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. SentMix-3L: A novel code-mixed test dataset in Bangla-English-Hindi for sentiment analysis. In *Proceedings of the First Workshop in South East Asian Language Processing*, pages 79–84, Nusa Dua, Bali, Indonesia. Association for Computational Linguistics.

Priya Rani, Gaurav Negi, Theodorus Fransen, and John P. McCrae. 2024a. Macms: Magahi code-mixed dataset for sentiment analysis.

Priya Rani, Gaurav Negi, Saroj Jha, Shardul Suryawanshi, Atul Kr. Ojha, Paul Buitelaar, and John P. McCrae. 2024b. Findings of the wildre shared task on code-mixed less-resourced sentiment analysis for indo-aryan languages. In *Proceedings of the 7th Workshop on Indian Language Data: Resources and Evaluation @LREC-COLING-2024 (WILDRE-7)*, Turin, Italy. ELRA Language Resources Association (ELRA) and the International Committee on Computational Linguistics (ICCL).

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2022. Don't stop fine-tuning: On training regimes for few-shot cross-lingual transfer with multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10725–10742, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN: Training deep convolutional neural network for Twitter sentiment classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 464–469, Denver, Colorado. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Gaurish Thakkar, Nives Mikelić Preradović, and Marko Tadić. 2021. Multi-task learning for cross-lingual sentiment analysis. In *Proceedings of the 2nd International Workshop on Cross-lingual Event-centric Open Analytics co-located with the 30th The Web Conference (WWW 2021), Ljubljana, Slovenia, April 12, 2021*, volume 2829 of *CEUR Workshop Proceedings*, pages 76–84. CEUR-WS.org.

Gaurish Thakkar, Nives Mikelić Preradović, and Marko Tadić. 2023. CroSentiNews 2.0: A Sentence-Level news sentiment corpus. In *Human Language Technologies as a Challenge for Computer Science and Linguistics - 2023*, pages 294–299, Poznan. Adam Mickiewicz University Press.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Manoel Veríssimo dos Santos Neto, Ayrton Amaral, Nádia Silva, and Anderson da Silva Soares.

2020. Deep learning Brasil - NLP at SemEval-2020 task 9: Sentiment analysis of code-mixed tweets using ensemble of language models. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1233–1238, Barcelona (online). International Committee for Computational Linguistics.

Mark Warschauer, Ghada R El Said, and Ayman G Zohry. 2002. Language choice online: Globalization and identity in Egypt. *Journal of Computer-Mediated Communication*, 7(4):744.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

MLInitiative@WILDRE7: Hybrid Approaches with Large Language Models for Enhanced Sentiment Analysis in Code-Switched and Code-Mixed Texts

Hariram Veeramani[1], Surendrabikram Thapa[2], Usman Naseem[3]
[1]UCLA, USA [2]Virginia Tech, USA [3]Macquarie University, Australia
[1]hariramveeramani@gmail.com, [2]sbt@vt.edu, [3]usman.naseem@mq.edu.au

Abstract

Code-switched and code-mixed languages are prevalent in multilingual societies, reflecting the complex interplay of cultures and languages in daily communication. Understanding the sentiment embedded in such texts is crucial for a range of applications, from improving social media analytics to enhancing customer feedback systems. Despite their significance, research in code-mixed and code-switched languages remains limited, particularly in less-resourced languages. This scarcity of research creates a gap in natural language processing (NLP) technologies, hindering their ability to accurately interpret the rich linguistic diversity of global communications. To bridge this gap, this paper presents a novel methodology for sentiment analysis in code-mixed and code-switched texts. Our approach combines the power of large language models (LLMs) and the versatility of the multilingual BERT (mBERT) framework to effectively process and analyze sentiments in multilingual data. By decomposing code-mixed texts into their constituent languages, employing mBERT for named entity recognition (NER) and sentiment label prediction, and integrating these insights into a decision-making LLM, we provide a comprehensive framework for understanding sentiment in complex linguistic contexts. Our system achieves competitive rank on all subtasks in the Code-mixed Less-Resourced Sentiment analysis (Code-mixed) shared task at WILDRE-7 (LREC-COLING).

Keywords: Code-switched language, Code-switched language, Sentiment analysis, Named entity recognition (NER), Large language models (LLMs)

1. Introduction

Informal communication constitutes a significant proportion of short text communications and online posts in our digital world (Tay, 1989). People tend to express themselves freely and spontaneously through various online platforms, ranging from social media to messaging apps. While some individuals stick to a single language when communicating, the use of two or more languages is also very common in informal communication. This phenomenon of code-mixing—mixing two or more languages within a single utterance—is common, especially in regions where closely related languages coexist (Thara and Poornachandran, 2018).

In code-mixing, individuals incorporate elements of different languages within their communication. This incorporation may occur for a multitude of reasons, including cultural affinity, linguistic convenience, or social dynamics (Lamabam and Chakma, 2016; Barman et al., 2014). For instance, individuals may switch between languages based on their proficiency, the context of the conversation, or the preferences of the people with whom they are conversing. By doing so, speakers can interact with ease and convey their intended messages more accurately. In non-English speaking and multilingual countries, code mixing is particularly prevalent due to the coexistence of multiple languages within the same socio-cultural space (Pratapa et al., 2018).

With a lot of code-mixed languages used, there is a need to automatically detect the sentiment of such code-mixed text important for various reasons (Kodali et al., 2022). Firstly, it allows for a deeper understanding of the emotions and opinions expressed by individuals in multilingual contexts. By accurately identifying sentiment, researchers and analysts can gain insights into the attitudes, preferences, and behaviors of diverse language communities. Secondly, sentiment analysis of code-mixed text enables businesses and organizations to effectively monitor and analyze customer feedback, social media trends, and public opinion in linguistically diverse markets. This information can inform marketing strategies, product development decisions, and customer relationship management efforts tailored to specific language communities (Joshi et al., 2016).

Moreover, sentiment analysis in code-mixed text can contribute to the development of more inclusive and culturally sensitive natural language processing (NLP) technologies (Chakravarthi et al., 2020). By recognizing and accounting for the linguistic nuances and variations present in multilingual communications, NLP models can better serve diverse user populations and facilitate more accurate language understanding and generation. Additionally, automatic sentiment detection in code-mixed text has implications for social and political analysis. By

WILDRE-7 Workshop, pages 66–72

analyzing sentiment patterns across different language groups, researchers can uncover insights into socio-political dynamics, cultural trends, and community sentiments, aiding in areas such as public policy formulation, cross-cultural communication, and conflict resolution.

In the seventh Workshop on Indian Language Data: Resources and Evaluation (WILDRE), a shared task on Code-mixed Less-Resourced Sentiment analysis was launched to address this issue. This shared task focuses on sentiment analysis in code-mixed data from less-resourced similar languages, particularly in language pairs and triplets of closely related Indo-Aryan languages spoken in eastern India. These languages include Magahi, Maithili, Bangla, and Hindi, along with English. The task aims to explore different machine learning and deep learning approaches to train models robust enough to perform well on the given training and validation datasets, thus providing insights into language representation and speakers' preferences in code-mixed settings. In this paper, we present our system description for this shared task. In our approach, we leverage named entity recognition, language decomposition, and large language models.

2. Related Works

The exploration of sentiment analysis in code-mixed text has been a subject of growing interest within the field of natural language processing (NLP), particularly due to the challenges and complexities associated with understanding and processing multilingual text. Several studies have laid the groundwork for addressing these challenges, providing valuable insights and methodologies for future research.

Pednekar and Saravanan (2023) addresses the scarcity of resources for sentiment analysis (SA) in mixed code languages by proposing the creation of a gold standard dataset. Their research aims to advance SA in underrepresented languages, highlighting the importance of high-quality datasets for evaluating SA models in languages with diverse code-mixing patterns (Pednekar and Saravanan, 2023).

Early work in the domain focused on identifying the occurrence and patterns of code-mixing across different linguistic contexts. A seminal study by Solorio and Liu (2008b) explored part-of-speech tagging for code-switched (a form of code-mixing) data. Their research highlighted the need for tailored NLP tools that can accurately process and understand the grammatical structures of mixed-language texts (Solorio and Liu, 2008a,b). Building upon these foundational insights, subsequent research has ventured into the sentiment analysis of code-mixed texts. For example, Joshi et al. (2016) developed algorithms that harness code-switching to improve sentiment analysis in bilingual text corpora. Their work underscored the potential benefits of leveraging the linguistic features inherent to code-switching for more nuanced sentiment detection.

In an effort to specifically address the challenges posed by code-mixed text in Indian languages, Barman et al. (2014) investigated code-mixing on Indian social media platforms. They created a corpus of code-mixed text and developed classification models that significantly improved the understanding of sentiment within these multilingual datasets.

The complexity of code-mixing and its implications for sentiment analysis have also been explored through competitions and shared tasks. For instance, the shared task on Sentiment Analysis for Dravidian Languages in Code-Mixed Text, as part of the Forum for Information Retrieval Evaluation (FIRE), has provided a platform for researchers to apply and evaluate various computational models on code-mixed datasets, leading to significant advancements in the field (Chakravarthi et al., 2020).

Similarly, Tho et al. (2020) provides a systematic literature review on code-mixed sentiment analysis using machine learning approaches. Their findings suggest that Support Vector Machine, Naïve Bayes, and Logistic Regression are the most common classifiers for this task, with Support Vector Machine exhibiting superior performance based on accuracy and F1 scores (Tho et al., 2020). Jin et al. (2023) offers a comprehensive review of text sentiment analysis methods and applications, exploring a variety of feature extraction and representation methods, including deep learning-based approaches. This review serves as a foundation for understanding the current status and development trends in SA (Jin et al., 2023). Similarly, Zucco et al. (2020) present a detailed study on sentiment analysis (SA) tools and methods for mining texts and social network data. Their analysis, based on objective criteria, highlights the importance of developing more advanced SA tools to enhance end-user experience (Zucco et al., 2020). Moreover, Habimana et al. (2020) review deep learning approaches for various SA tasks, suggesting that the future of SA models could benefit from incorporating advanced techniques such as BERT, sentiment-specific word embedding models, and attention mechanisms (Habimana et al., 2020).

3. Task Descriptions

We only participate in the first shared task, i.e. Code-mixed less-resourced sentiment analysis. This shared task aimed to address the complexities of sentiment analysis in code-mixed data from less-resourced similar languages, with a focus

on Magahi-Hindi-English, Maithili-Hindi, Bangla-English-Hindi, and Hindi-English language pairs and triplets. These languages, belonging to the Indo-Aryan language family and predominantly spoken in eastern India, present unique challenges due to their linguistic similarities and low-resource settings. An important aspect of this shared task was the introduction of an unlabelled test dataset for the code-mixed Maithili language (Maithili-Hindi-English) (Rani et al., 2024b). Participants were challenged to leverage the available training datasets from Magahi-Hindi-English, Maithili-Hindi, Bangla-English-Hindi, and Hindi-English to determine the sentiment of comments in this target language.

Participants were tasked with exploring different machine learning and deep learning approaches to train models on the training and validation data sets provided. The goal was to develop models robust enough to perform well on code-mixed language datasets, thus enhancing sentiment analysis capabilities in multilingual contexts.

3.1. Evaluation

The shared task on CodaLab employed standard evaluation metrics, primarily the average F1 score, to assess participating teams' models. The evaluation also included precision, recall, and F1 scores across sentiment classes for detailed analysis. Initially, teams accessed training and validation data, with test data and the Maithili test set later released. Two tracks were defined: one for determining polarity in code-mixed settings and another for sentiment analysis in code-mixed Maithili. Datasets were divided into train, validation, and test sets, with a 70:15:15 ratio. For the combined language pairs, training and validation sets were merged. Submitted models were evaluated based on their ability to predict sentiment labels on test data. Results, including F1 scores, precision, and recall, were provided to teams for analysis and discussion, offering insights into code-mixed sentiment analysis challenges and solutions.

4. Dataset

The dataset provided for the shared task comprised annotated code-mixed text in three language pairs: Magahi-Hindi-English, Bangla-English-Hindi, and Hindi-English. Each comment or sentence in the Magahi-Hindi-English and Hindi-English datasets was labeled with four sentiment categories: Positive, Negative, Neutral, or Mixed (Rani et al., 2024a). In contrast, the Bangla-English-Hindi dataset was labeled with three sentiment categories: Positive, Negative, or Neutral.

The Magahi-Hindi-English and Hindi-English datasets were collected from various YouTube channels and meticulously annotated with the assistance of native speakers of the respective languages. This ensured that the data accurately reflected the nuances of sentiment expression in these code-mixed contexts. Additionally, for the Bangla-English-Hindi dataset, the SentMix-3L dataset by Raihan et al. (2023) was utilized. This dataset provided a rich collection of code-mixed text in Bangla, English, and Hindi, offering valuable insights into sentiment analysis in a multilingual context.

Participants in the shared task were allowed to leverage external resources, provided they were openly available and could be used by other participants for research purposes. Proper citation and detailed information about any external dataset utilized were included in the system description paper submitted by participants. Overall, the dataset offered a diverse collection of code-mixed text across different language pairs and sentiment categories, enabling participants to develop and evaluate models robust enough to handle sentiment analysis in code-mixed data from less-resourced similar languages.

5. System Description

Our system for sentiment analysis in code-mixed texts employs a multi-step approach, leveraging the capabilities of large language models (LLMs) and the multilingual BERT (mBERT) model to accurately process and analyze sentiment in less-resourced, code-mixed language data. As shown in Figure 1, our methodology consists of the following steps:

5.1. Decomposition of Code-Mixed Language into Individual Languages

The first step in our approach involves decomposing the code-mixed language data into its constituent languages. This process is crucial for handling the intricacies of code-mixed texts and allows for more accurate subsequent analysis. We utilize three prominent LLMs: Mistral, LLama (Touvron et al., 2023), and Gemma, to perform this decomposition. By prompting these models with the specific languages present in the code-mixed text, as illustrated in Figure 1, we effectively separate Hindi-English code-mixed language into individual Hindi and English components. This decomposition aids in the further processing and understanding of the text.

5.2. Finetuning mBERT

We use mBERT for two major objectives: Named Entity Recognition (NER) and label prediction.

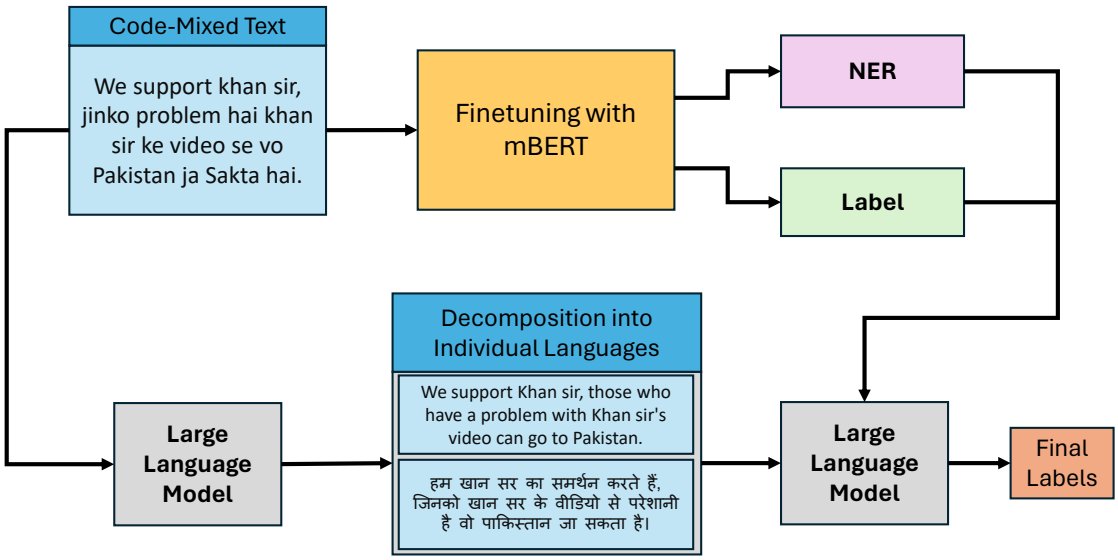


Figure 1: System Description of Our Approach

5.2.1. Use of mBERT for NER

For NER: We employ mBERT (Devlin et al., 2019), a model pre-trained on multiple languages, to perform NER on the decomposed language texts. NER is instrumental in identifying key entities within the text, providing valuable context that enhances the sentiment analysis process (Li et al., 2020). mBERT's ability to understand multiple languages makes it particularly suited for this task, enabling accurate entity recognition in all language components of the code-mixed text.

5.2.2. For Label Prediction

Additionally, we leverage mBERT for the prediction of sentiment labels in the test dataset. By fine-tuning mBERT with our training data, we are able to classify the sentiment of code-mixed texts effectively. The fine-tuned mBERT model is then used for inference on the test data, predicting the sentiment labels with a high degree of accuracy.

5.3. Large Language Models for Final Decision

In the final step of our approach, we integrate the outputs from the previous steps—including the NER results, sentiment labels from mBERT, and the decomposed language components—into a comprehensive input for a large language model. This LLM is tasked with making the final sentiment analysis decision. By providing the LLM with a holistic view of the text, including both the original code-mixed form and the derived insights from mBERT and language decomposition, we enable it to leverage all available information for conflict resolution and final sentiment classification. This step is crucial for resolving any discrepancies between the sentiment labels predicted by mBERT and the nuances captured through NER and language decomposition, ensuring a cohesive and accurate sentiment analysis outcome.

This multi-step approach leverages the complementary strengths of LLMs and mBERT, facilitating a nuanced and effective analysis of sentiment in code-mixed texts, particularly in the context of less-resourced languages. Through this methodology, we address the challenges posed by code-mixing and provide insights into the sentiments expressed in multilingual communities.

6. Results

Our participation in the first shared task—Code-mixed less-resourced sentiment analysis—yielded notable results across various language combinations, including Bangla-English, Hindi-English, Magahi-Hindi-English, and Maithili-Hindi-English. We evaluated our model's performance using three distinct Large Language Models (LLMs): Mistral, Llama, and Gemma, across the criteria of macro-averaged F1 score, precision, and recall. Table 1 summarizes our findings.

In the Bangla-English combination, Mistral outperformed its counterparts with a macro-averaged F1 score of 0.67, precision of 0.76, and recall of 0.68. This result indicates Mistral's superior capability in handling the intricacies of Bangla-English code-mixed texts. For the Hindi-English dataset, Gemma led with a macro-averaged F1 score of

Language Combination	LLM	Macro-Averaged F1	Macro-Averaged Precision	Macro-Averaged Recall
Bangla-English	Mistral	**0.67**	**0.76**	**0.68**
	Llama	0.34	0.58	0.41
	Gemma	0.64	0.66	0.63
Hindi-English	Mistral	0.31	**0.38**	**0.47**
	Llama	0.28	0.36	0.32
	Gemma	**0.34**	0.35	0.39
Magahi-Hindi-English	Mistral	0.23	**0.39**	**0.39**
	Llama	0.21	0.29	0.19
	Gemma	**0.26**	0.28	0.27
Combined*	Mistral	0.33	**0.40**	**0.38**
	Llama	0.26	0.33	0.28
	Gemma	**0.35**	0.36	0.36
Maithili-Hindi-English	Mistral	0.13	0.26	0.27
	Llama	**0.35**	**0.36**	**0.36**
	Gemma	0.24	0.26	0.31

Table 1: Performance of our model with different datasets. *The combined language represents Bangla-English, Hindi-English and Magahi-Hindi-English datasets altogether.

0.34, albeit Mistral showed better performance in terms of precision and recall. This suggests that while Gemma was more effective overall, Mistral was better at identifying relevant instances, albeit with a higher rate of false positives.

Magahi-Hindi-English texts, which represent a more challenging setting due to their triple-language mix, saw Gemma performing the best in terms of the F1 score. However, Mistral consistently showed higher precision and recall, indicating its effectiveness in accurately classifying sentiments in this complex language mix. When evaluating the combined dataset, which includes all language pairs, Gemma again demonstrated the highest F1 score, highlighting its robustness across multiple code-mixed settings. Mistral, however, maintained higher precision and recall scores, reinforcing its efficiency in identifying sentiment with greater accuracy.

Notably, the Maithili-Hindi-English combination presented a unique challenge. In this case, Llama achieved the best performance across all metrics, underscoring its effectiveness in dealing with the code-mixed Maithili language. This performance emphasizes the potential of LLMs in addressing sentiment analysis in less-explored language combinations. These results highlight the ability of our approach in leveraging LLMs for sentiment analysis in code-mixed texts. The varied performance across different models and language combinations shows the importance of model selection based on the specific linguistic characteristics of the target data. Our findings contribute to the broader understanding of sentiment analysis in multilingual contexts, especially within less-resourced languages.

7. Conclusion

In conclusion, our exploration of sentiment analysis in code-mixed and code-switched texts across less-resourced languages demonstrates the significant potential of leveraging Large Language Models (LLMs) such as Mistral, Llama, and Gemma. Our methodology, which intricately combines language decomposition, named entity recognition (NER), and sentiment classification, showcases a novel approach to navigating the complexities inherent in multilingual sentiment analysis. The results across various language combinations underscore the importance of model and technique selection tailored to the specific challenges posed by each language mix. Through this work, we not only contribute to the understanding of sentiment analysis in the context of code-mixed and code-switched languages but also highlight the importance of developing NLP tools that are inclusive of linguistic diversity. Our findings pave the way for future research to further refine these methods and expand the scope of sentiment analysis in multilingual and multicultural societies, ensuring that NLP technologies remain responsive to the nuances of human language and emotion.

References

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.

Bharathi Raja Chakravarthi, Ruba Priyadharshini,

Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P McCrae. 2020. 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*, pages 21–24.

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.

Olivier Habimana, Yuhua Li, Ruixuan Li, Xiwu Gu, and Ge Yu. 2020. Sentiment analysis using deep learning approaches: an overview. *Science China Information Sciences*, 63:1–36.

Yuxin Jin, Kui Cheng, Xinjie Wang, and Lecai Cai. 2023. A review of text sentiment analysis methods and applications. *Frontiers in Business, Economics and Management*, 10(1):58–64.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word 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.

Prashant Kodali, Anmol Goel, Monojit Choudhury, Manish Shrivastava, and Ponnurangam Kumaraguru. 2022. Symcom-syntactic measure of code mixing a study of english-hindi code-mixing. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 472–480.

Priyadarshini Lamabam and Kunal Chakma. 2016. A language identification system for code-mixed english-manipuri social media text. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 79–83. IEEE.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Chaitanya B Pednekar and P Saravanan. 2023. A study on different methods in sentiment analysis from text. In *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1115–1122. IEEE.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.

Md Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anstasopoulos, and Marcos Zampieri. 2023. Sentmix-3l: A bangla-english-hindi code-mixed dataset for sentiment analysis. *arXiv preprint arXiv:2310.18023*.

Priya Rani, Gaurav Negi, Theodorus Fransen, and John P. McCrae. 2024a. Macms: Magahi code-mixed dataset for sentiment analysis. *arXiv preprint arXiv:2403.04639*.

Priya Rani, Gaurav Negi, Saroj Jha, Shardul Suryawanshi, Atul Kr. Ojha, Paul Buitelaar, and John P. McCrae. 2024b. Findings of the wildre shared task on code-mixed less-resourced sentiment analysis for indo-aryan languages. In *Proceedings of the 7th Workshop on Indian Language Data: Resources and Evaluation @LREC-COLING-2024 (WILDRE-7)*, Turin, Italy. ELRA Language Resources Association (ELRA) and the International Committee on Computational Linguistics (ICCL).

Thamar Solorio and Yang Liu. 2008a. Learning to predict code-switching points. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 973–981.

Thamar Solorio and Yang Liu. 2008b. Part-of-speech tagging for english-spanish code-switched text. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1060.

Mary WJ Tay. 1989. Code switching and code mixing as a communicative strategy in multilingual discourse. *World Englishes*, 8(3):407–417.

S Thara and Prabaharan Poornachandran. 2018. Code-mixing: A brief survey. In *2018 International conference on advances in computing, communications and informatics (ICACCI)*, pages 2382–2388. IEEE.

Cuk Tho, Harco Leslie Hendric Spits Warnars, Benfano Soewito, and Ford Lumban Gaol. 2020. Code-mixed sentiment analysis using machine learning approach–a systematic literature review. In *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–6. IEEE.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open

and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Chiara Zucco, Barbara Calabrese, Giuseppe Agapito, Pietro H Guzzi, and Mario Cannataro. 2020. Sentiment analysis for mining texts and social networks data: Methods and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(1):e1333.

Aalamaram: A Large-Scale Linguistically Annotated Treebank for the Tamil Language

A M Abirami[1*], Wei Qi Leong[2,3*], Hamsawardhini Rengarajan[2,3*], D Anitha[1], R Suganya[4], Himanshu Singh[5], Kengatharaiyer Sarveswaran[6,7], William Chandra Tjhi[2,3], Rajiv Ratn Shah[5]

[1]Thiagarajar College of Engineering, Madurai, India
{abiramiam,anithad}@tce.edu

[2]AI Singapore, Singapore
[3]National University of Singapore, Singapore
{weiqi,hamsa,wtjhi}@aisingapore.org

[4]Vellore Institute of Technology, Chennai, India
suganya.ramamoorthy@vit.ac.in

[5]Indraprastha Institute of Information Technology, Delhi, India
{himanshu17291,rajivratn}@iiitd.ac.in

[6]University of Jaffna, Sri Lanka
[7]University of Konstanz, Germany
sarves@univ.jfn.ac.lk

Abstract

Tamil is a relatively low-resource language in the field of Natural Language Processing (NLP). Recent years have seen a growth in Tamil NLP datasets in Natural Language Understanding (NLU) or Natural Language Generation (NLG) tasks, but high-quality linguistic resources remain scarce. In order to alleviate this gap in resources, this paper introduces *Aalamaram*, a treebank with rich linguistic annotations for the Tamil language. It is hitherto the largest publicly available Tamil treebank with almost 10,000 sentences from diverse sources and is annotated for the tasks of Part-of-speech (POS) tagging, Named Entity Recognition (NER), Morphological Parsing and Dependency Parsing. Close attention has also been paid to multi-word segmentation, especially in the context of Tamil clitics. Although the treebank is based largely on the Universal Dependencies (UD) specifications, significant effort has been made to adjust the annotation rules according to the idiosyncrasies and complexities of the Tamil language, thereby providing a valuable resource for linguistic research and NLP developments.

Keywords: Tamil Corpus, CoNLL-U, Annotation Guidelines, Tamil Treebank, Universal Dependencies

1. Introduction

Tamil, with a rich literary tradition spanning over two millennia, stands as one of the oldest surviving classical languages globally. Officially recognized by the Indian government as a classical language in 2004, Tamil holds significant cultural and historical importance, extending beyond being merely a means of communication (Keane, 2004). Boasting a global speaker base of approximately 89.7 million people[1], Tamil's influence is not only confined to its native regions such as India and Sri Lanka, but also extends to diaspora communities in countries like Singapore, Malaysia, Mauritius, Fiji, and South Africa[2].

However, despite the relatively large population that uses the language, the amount of data available for Natural Language Processing (NLP) in Tamil is arguably not commensurate, lagging behind major languages such as English, French, Spanish and Chinese. Although recent years have seen a growth in unannotated Tamil corpora (Kunchukuttan et al., 2020; Kakwani et al., 2020; Ramesh et al., 2021) as well as annotated data for certain benchmarking tasks in Natural Language Understanding (NLU) and Natural Language Generation (NLG) (Kakwani et al., 2020; Doddapaneni et al., 2023), datasets with rich linguistic annotations remain scarce.

Such annotated corpora, commonly known as treebanks, are important sources of data not just for linguistic research, but also for practical applications in NLP. Syntactic parse trees can be used directly in grammar checking (Li et al., 2022) and linguistic features engineered via syntactic parsers

*Co-first authors

[1]https://www.worlddata.info/languages/tamil.php

[2]https://www.britannica.com/topic/Tamil-language

WILDRE-7 Workshop, pages 73–83

can be used to enrich text representations and improve performance of models on downstream tasks such as machine translation (Deguchi et al., 2019; Bugliarello and Okazaki, 2020), machine reading comprehension (Zhang et al., 2020), and text summarization (Xu and Durrett, 2019; Huang et al., 2022). Moreover, although Large Language Models (LLMs) generally display strong performance in these aforementioned tasks, they still have room for improvement when it comes to understanding the correct morphosyntax of languages (Zhou et al., 2023), especially for low-resource languages such as Tamil (Leong et al., 2023), and preliminary research has shown that this gap can potentially be closed with treebanks as well (Yoshida et al., 2024). As such, it would be important to have treebanks built for the Tamil language as well in order to push the envelope of Tamil NLP systems.

As of now, there are two publicly available Tamil treebanks built under the Universal Dependencies (UD) framework (Nivre et al., 2016) – the Tamil Treebank (TTB) (Ramasamy and Žabokrtský, 2012) and the Modern Written Tamil Treebank (MWTT) (Krishnamurthy and Sarveswaran, 2021). Unfortunately, both treebanks are rather small, with a size of approximately 600 sentences each (see Table 1), which is not ideal for the training of end-to-end deep neural networks. Furthermore, these treebanks are also highly limited in data diversity, being drawn only from a single data source. This could reduce the effectiveness of models trained on them as they might not be able to generalize beyond the sentence structures and domains present in these treebanks. These treebanks also lack named entity annotations, which are important for information extraction applications.

As such, we propose *Aalamaram*[3], a large-scale treebank with almost 10,000 Tamil sentences annotated for parts-of-speech (POS), morphological features, named entities and dependency relations (see Figure 1). It is hitherto the largest publicly-available treebank for the Tamil language. *Aalamaram* is built from diverse data sources and significant efforts have been made to review and adjust the annotation rules from the UD framework and past Tamil treebanks in order to account for the idiosyncrasies and complexities of the Tamil language.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 describes the data curation process in detail. Section 4 dives into the annotation process and quality control cycle. Section 5 discusses certain linguistic phenomena that surfaced during the annotation process and which prompted reanalysis. Finally, we present our conclusions in Section 6 and put forward suggestions for future works.

2. Related Work

Although Tamil is a relatively low resource language, it is still classified as a class 3 language[4] according to Joshi et al.'s (2020) taxonomy, and this is possibly in part a result of the growth in raw Tamil text corpora for unsupervised pre-training in recent years, such as IndicNLP (Kunchukuttan et al., 2020) and IndicCorp (Kakwani et al., 2020). In addition, there have also been parallel efforts in building annotated datasets for certain tasks in NLU and NLG such as machine translation (Ram R and Devi, 2018; Siripragada et al., 2020; Ramesh et al., 2021), question answering and sentiment analysis (Doddapaneni et al., 2023).

However, these datasets often lack the linguistic annotations that are essential for a granular syntactic and semantic analysis of Tamil texts. Such detailed analyses are vital in facilitating downstream NLP applications that require a nuanced understanding of the language. Currently, there have been a couple of efforts that looked at building such specialized corpora, tackling tasks such as POS tagging (Dhanalakshmi et al., 2009; Akilan and Naganathan, 2012; Chandra et al., 2014; Devi et al., 2016; Sarveswaran and Dias, 2021) and Named Entity Recognition (NER) (Pattabhi and Devi, 2013; Mhaske et al., 2023). However, there is a lack of a unified tag set for these linguistic annotations, which can make it difficult to harmonize and pool resources as well as to compare results across studies.

One promising work in unifying morphosyntactic annotations not just intra-linguistically but also cross-linguistically is the UD framework (Nivre et al., 2016). It aims to provide a linguistic representation conducive for morphosyntactic research, semantic interpretation, as well as practical NLP across diverse human languages (de Marneffe et al., 2021).

There have been to date two seminal works in applying the UD framework to the Tamil language, namely the Tamil Treebank (TTB) (Ramasamy and Žabokrtský, 2012) and the Modern

[3] *Aalamaram* (ஆலமரம்) is the Tamil word for the banyan tree, which is culturally significant to Tamilians. It is often featured in Tamil literature, folklore and proverbs, signifying its deep-rooted presence within the Tamil community. The use of the name *Aalamaram* is also a direct reference to the fact that the resource built is a treebank containing parse trees.

[4] Joshi et al. (2020) classified languages based on their existing resources into 6 categories, with class 3 languages being referred to as "rising stars" which have unsupervised pre-training data but lack labeled data collection.

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC
# sent_id = ebooks_719_F40BAF8E_0									
# sent_no = 259									
# text = மர்தானா கண்களைத் திறந்தான்.									
# text_en = Mardana opened his eyes.									
# translit = martāṉā kaṇkaḷait tiṟantāṉ.									
# source = punjabikathaigal_ebooks_project_madurai									
1	மர்தானா	மர்தானா	PROPN	PROPN	Animacy=Hum|Case=Nom|Gender=Masc|Number=Sing	3	nsubj	_	Entity=B_INDIV
2	கண்களைத்	கண்	NOUN	NOUN	Animacy=Nhum|Case=Acc|Gender=Neut|Number=Plur	3	obj	_	_
3	திறந்தான்	திற	VERB	TR	Animacy=Hum|Gender=Masc|Mood=Ind|Polarity=Pos|Tense=Past|VerbForm=Fin|Voice=Act	0	root	_	SpaceAfter=No
4	.	.	PUNCT	PUNCT	PunctType=Peri	3	punct	_	_

Figure 1: Example of an annotated sentence in *Aalamaram*

	TTB	MWTT	Aalamaram
Sentences	600	534	9567
Tokens	8635	2536	84253
Syntactic Words	9581	2584	95384
Multi-word Tokens	835	43	10211
Syntactic Word to Multi-word Token Ratio	2.13	2.12	2.09

Table 1: Comparison of Existing Tamil Treebanks with *Aalamaram*

Written Tamil Treebank (MWTT) (Krishnamurthy and Sarveswaran, 2021). TTB contains 600 sentences of news data and was initially annotated according to the Prague Dependency Treebank scheme (Hajič, 1998; Hajič et al., 2020) with 3 layers of annotations, including a morphological layer, surface syntax layer, and a tectogrammatical layer. It was then subsequently converted into the UD format. MWTT on the other hand contains 534 simple sentences sourced from Thomas Lehmann's reference grammar for the Tamil language "A Grammar of Modern Tamil" (Lehmann, 1993). MWTT was created with the intention of providing an error-free gold standard benchmark treebank for Tamil through the coverage of different sentence structures provided in the reference grammar, as it was observed that there were certain inconsistencies and errors in TTB that might have been a result of the automatic mapping from the Prague Dependency Treebank format to the UD format.

While both treebanks have been important resources given the dearth of morphosyntactically annotated datasets in Tamil, they are both relatively small and not ideal for the training of end-to-end neural networks. In fact, MWTT was also intended to be used only as a test dataset. Furthermore, they are both limited in the domains that are covered, with MWTT being drawn from a reference grammar and TTB being drawn from news only.

In addition, the highly agglutinative nature of Tamil (Lehmann, 1993; Krishnamurti, 2003; Annamalai and Steever, 2019) poses a challenge in determining the appropriate tokenization of Tamil words. A case in point would be the widespread occurrence of clitics which serve a gamut of semantic and pragmatic functions (Lehmann, 1993; Schiffman, 1999; Annamalai and Steever, 2019). These clitics are only marginally dealt with in the two existing Tamil UD treebanks, but a more in-depth treatment of the matter would be crucial in ensuring accurate analysis of Tamil texts. Both treebanks are also not annotated for named entities which are important in information extraction applications. As such, there is a need for a larger Tamil treebank with diverse data sources to support the training of deep neural networks, with named entity annotations to support NER applications, as well as a need for in-depth analysis of various linguistic phenomena in the Tamil language in order to arrive at a more accurate annotation. We therefore propose *Aalamaram* as a new treebank for the Tamil language in order to plug this gap.

3. Data Curation

As previous treebanks were relatively small and/or limited to a single source, we wanted to create a treebank that was larger in scale, with greater variety in data sources, and that also contained named entity annotations. Comparative statistical analysis of the Tamil treebanks is presented in Table 1, highlighting the growth in dataset scale in the proposed *Aalamaram* treebank. We also wanted

the data to reflect real-world usage of Tamil, albeit with a focus on formal language for a start. This section describes the process of collecting and curating the data to arrive at the final set of sentences for annotation.

3.1. Data Sources

In order to enrich the diversity of texts in our dataset, we extracted data from a variety of sources:

- News - News articles written between 2021 and 2022 were scraped from Theekkathir[5], a Tamil newspaper operated by the Toiling Masses Welfare Trust Tamil Nadu. The data scraped primarily comprises formal news articles, with a predominant focus on political affairs. This data is available under a CC-BY-SA 4.0 IN license.
- Movie Reviews - Movie reviews were sourced from an existing dataset[6]. The language used in this source is not as formal as in the other sources.
- Wikipedia - Wikipedia articles were sourced from an existing dataset[7] and additional scraping of Wikipedia was done in order to enrich the representation of named entities in the dataset.
- Ebooks - Ebooks spanning publication dates from 1900 to 2021 were obtained from the Free Tamil Ebooks website[8]. These comprise mostly novels and are in the domain of fiction. These ebooks are mostly licensed under a CC-BY-NC-SA 4.0 license.
- Grammar books - Simple sentences were collected from Indian middle and high school Tamil grammar books, as they encompassed relatively simple examples that are well-crafted to demonstrate a variety of grammar rules. These sentences were only used in the initial phase for training the annotators, as well as for developing the annotation guidelines.

3.2. Data Filtering

After extracting all the data from the various sources, a series of data filtering steps were taken in order to obtain a subset that is suitable for linguistic annotation.

Although the goal for this initial work is to annotate approximately 10,000 sentences on sentence-level tasks, we initially filtered data on a paragraph level in order to obtain a set of paragraphs that could be used for paragraph-level or discourse annotations in the future. The final set of sentences were samples from this set of paragraphs. The following were the exclusion criteria that we set for removing data from the pool:

- Paragraph consists of more than 4 sentences
- 50% or more of the words in the paragraph are English
- High frequency of numerals are present in the paragraph
- Paragraphs begin with certain symbols such as , or !
- Sentences in the paragraph are shorter than 3 words or longer than 30 words

This allowed us to balance filtering out undesirable content and retaining useful data, with approximately 30% of the data being removed after these steps.

3.3. Sampling Strategy

The next step was to obtain a set of approximately 10,000 sentences from the pool of paragraphs from the filtering stage. We performed stratified random sampling by data source to obtain a corpus of 7,900 paragraphs with 30,000 sentences which can be used for future paragraph-level annotations. The target ratio of data sources (30% Wikipedia, 20% News, 20% Movie Reviews and 30% Ebooks) was decided through practical considerations of data availability as well as balance between sources.

The final set of sentences were filtered via another round of stratified random sampling with the same target ratios, with sentence segmentation performed using punctuation as boundaries. Upon inspection of the data, it was found that using punctuation for sentence segmentation may occasionally result in incomplete sentences. As such, we merged such split sentences back into a single sentence as far as possible and purged malformed ones that could not be salvaged from the dataset. This resulted in a final set of 9567 sentences available for linguistic annotation (see Table 2 for statistics).

4. Data Annotation

4.1. Annotation and Quality Control

The annotation process was divided into 3 main phases – Guideline Development Phase, Training

[5] https://theekkathir.in/

[6] https://www.kaggle.com/datasets/sudalairajkumar/tamil-nlp

[7] https://www.kaggle.com/datasets/disisbig/tamil-wikipedia-articles

[8] https://freetamilebooks.com/

Data Source	Sentences	Tokens	Syntactic Words	Multi-word Tokens	Proportion (Sentences)
News	1717	14959	17140	1954	17.95%
Movie Reviews	2191	22262	25054	2615	22.90%
Wikipedia	3098	29751	33319	3288	32.38%
Ebooks	2561	17281	19871	2354	26.77%
Total	9567	84253	95384	10211	100.00%

Table 2: Statistics of Various Data Sources in *Aalamaram*

Phase, and Annotation Phase. A total of 20 undergraduate and postgraduate students who are native speakers of Tamil and who are majoring in Data Science and Information Technology were recruited for this project. The quality control team involved 3 professors and 4 postgraduate students studying Data Science who also have Tamil as their native language and who have experience in NLP.

In the first phase, guidelines were developed with a top-down approach, using the UD guidelines as the main reference and drawing further inspiration from existing NER datasets (Sekine et al., 2002; Vijayakrishna and Sobha, 2008; Weischedel et al., 2011) and Tamil datasets with linguistic annotations. The guidelines were then further refined based on iterative linguistic analyses.

In the second phase, annotators were trained on the annotation guidelines using 200 sentences from grammar books as practice. The 20 annotators were divided into two teams of 10 (named Team 1 and Team 2). They were then further divided into 5 pairs each, one pair for each annotation task, namely POS, Lemma, Morphology, Dependency Relations, and NER. A careful learning and review process was put in place in which each member of a pair would review every annotation done by the other member. This allowed the annotators to reinforce their understanding of the guidelines and to surface challenges in regular discussions with quality controllers. Grammar book sentences were chosen for this phase as they are relatively more straightforward and helped to get annotators up to speed quickly without being bogged down unnecessarily by complicated cases. This phase also allowed us to update the guidelines based on feedback from the annotators' experiences. Inter-annotator agreement (IAA) scores based on Cohen's kappa score (Berry and Mielke, 1988) were also calculated at regular intervals to monitor the annotators' performance and the quality of the annotation.

Finally, following verification by the quality control team to ascertain the readiness of the annotators, determined through a combination of regular assessments and IAA scores, the annotators proceeded to work on the actual dataset consisting of 9567 sentences in the Annotation Phase. This phase was done without cross-reviews between members of each pair in order to speed up annotation. Team 1 and Team 2 were also not allowed to view each other's annotations to avoid inadvertent biases in annotation. 10% of the dataset selected at random was annotated by both Team 1 and 2 to allow for calculation of IAA scores. Simultaneously, this same set of sentences was also annotated by the quality control team and termed the "Gold" dataset. This allowed us to calculate the IAA between the two teams and the quality controllers in order to ascertain the accuracy of the annotations. The IAA reaches or exceeds 0.7 between both teams as well as between teams and the quality controllers (see Table 3), which indicates substantial agreement. We do not include the IAA for named entity annotation at the moment as reviews are still in progress. Furthermore, we also observed significant improvements in IAA between the initial and final stages of annotation (see Figure 2), suggesting that the quality control cycle was effective in improving the dataset quality over time. At the end of this phase, the data underwent a final quality check as well as automatic validation using the UD script[9]. This process resulted in some updates to the rules included in the UD for Tamil treebanks, showing the success of our large-scale treebank in expanding the variety of linguistic phenomena covered.

4.2. Annotation Tasks

4.2.1. Multi-word Segmentation

Given the highly agglutinative nature of Tamil, we decided to pay close attention to how words should be tokenized to best capture morpho-syntactic information in our dataset. We split auxiliary verbs and postpositions out as separate tokens, which is in line with existing work (Ramasamy and Žabokrtský, 2012; Krishnamurthy and Sarveswaran, 2021). Furthermore, we also split all clitics as listed in Lehmann (1993), which

[9] https://github.com/UniversalDependencies/tools/blob/master/validate.py

	UPOS	XPOS	HEAD	DEPREL
Team 1 vs Gold	0.8594	0.8185	0.7293	0.7003
Team 2 vs Gold	0.8748	0.8311	0.8081	0.7747
Team 1 vs Team 2	0.8342	0.7941	0.7275	0.6997

Table 3: Inter-annotator Agreement Scores for Full Dataset Annotation

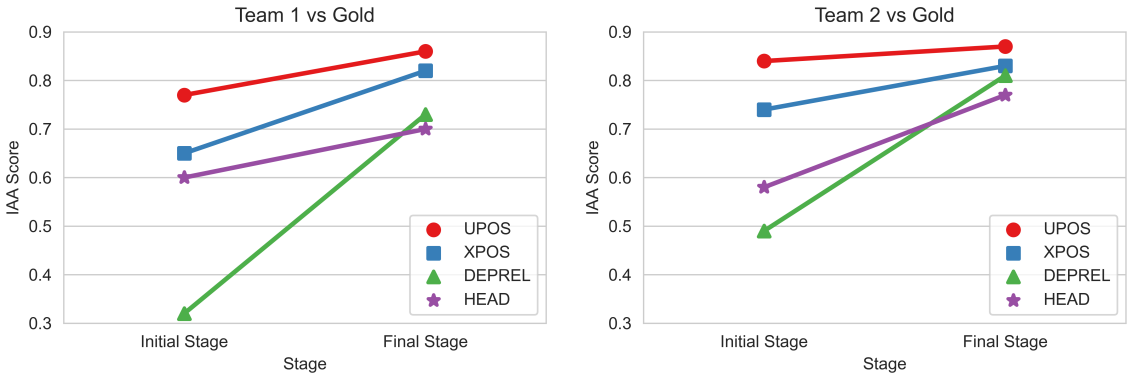


Figure 2: Improvement in Inter-annotator Agreement Scores

generally lack coverage in existing treebanks[10]. This allows us to better clarify the function of these clitics (see Section 5.2) in these sentences. On the other hand, we eschew the tokenization of case markers (as is done in TTB) and instead opt to acknowledge them under morphological feature annotations which is more in line with the UD annotation guidelines. We also do not split compound nouns.

4.2.2. POS Annotation

For POS annotations, we include both the Universal POS (UPOS) and more fine-grained language-specific (XPOS) tags. All 17 UPOS tags of the UD are used in *Aalamaram*, in contrast to TTB and MWTT which lack `SCONJ`, `INTJ` and `SYM`. This can be attributed to the scale and the coverage of *Aalamaram*. Certain words such as *eṉpatu*, and clitics such as *-um* were also re-analyzed in certain contexts as `SCONJ` (see Section 5.2), contributing to this difference.

4.2.3. Morphological Feature Annotation

The agglutinative nature of Tamil morphology makes the accurate analysis of morphological features crucial in NLP applications. As such, *Aalamaram* uses an expanded set of features compared to MWTT and TTB. One example of this expansion is in the annotation of the Animacy feature.

In MWTT and TTB, only the `Anim` label for animate nouns is used. However, nouns in Tamil have been analyzed in linguistic literature as being grouped along the axis of *rationality*[11] (Lehmann, 1993; Annamalai and Steever, 2019). Rational nouns include human-like entities such as humans, gods and demons, while irrational nouns can include both animate nouns like animals and babies as well as inanimate nouns. Rationality has a significant impact on grammar, such as in determining the inflection of nouns in certain grammatical cases or in subject-verb agreement. Although preliminary research has suggested that there can be intersections between rationality and animacy, with certain word inflections dependent on one but not the other[12], we leave exploration of this intersection to future work and tentatively use `Hum` and

[10]MWTT does not tokenize clitics and TTB only covers 4 clitics, namely *-um*, *-ē*, *-ēyē*, and *-āvatu*.

[11]This is sometimes referred to as [±human] (Krishnamurti, 2003), with rational nouns called உயர்திணை (*uyartiṇai*) and irrational nouns called அஃறிணை (*aḫṟiṇai*) in the Tolkāppiyam.

[12]For example, in the sentence *kumār ūr-ukkup pōṉāṉ* (Kumar went to a town), the inanimate noun *ūr* (town) takes the dative case marker *-ukku*. On the other hand, in a similar sentence like *kumār āppāv-iṭam pōṉāṉ* (Kumar went to father), the word *āppā* (father) has to take the locative case marker *-iṭam* instead due to it being an animate noun (Lehmann, 1993). This variation seems to be dependent on animacy and not on rationality, since the irrational animate noun *kuḻantai* (baby) takes the locative case marker *-iṭam* as well.

Nhum for the Animacy values, with Hum being used for rational nouns and Nhum for irrational nouns.

4.2.4. Dependency Relation Annotation

The dependency relations in *Aalamaram* were also annotated according to the UD guidelines, using 28 out of 37 relations, which is an expansion from the 22 used in MWTT and 25 in TTB. Significant linguistic inquiry was carried out in order to derive accurate dependency relations, especially due to the more extensive multi-word segmentation that was carried out. Some of these are explored in Section 5.

4.2.5. Named Entity Annotation

For named entities, we designed a hierarchical tagset with three levels of granularity, drawing inspiration from existing named entity hierarchies (Sekine et al., 2002; Vijayakrishna and Sobha, 2008) as well as the OntoNotes NER tagset (Weischedel et al., 2011). The first level comprises the standard ENAMEX, NUMEX and TIMEX labels, while the second and third levels comprise 14 and 35 fine-grained tags respectively. We also follow common conventions in employing the IOB2 tagging scheme.

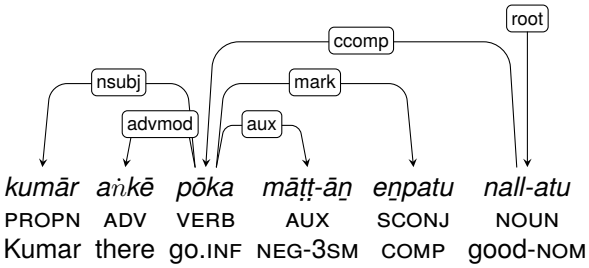
5. Discussion

This section discusses some of the linguistic phenomena in Tamil that surfaced through the annotation process and which prompted reanalysis.

5.1. *eṉpatu*

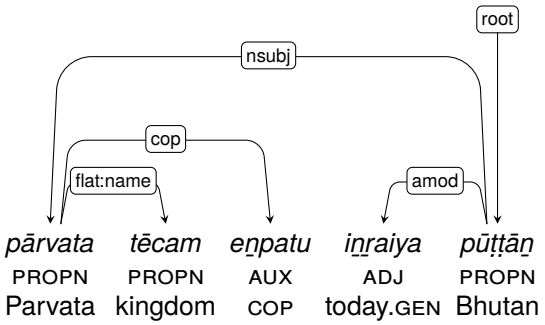
The word *eṉpatu*, which is the future verbal noun form of the verb *eṉ* (to say), has traditionally been analyzed as a complementizer (Lehmann, 1993), which would fall under the label of SCONJ (subordinating conjunction) under the UD framework, although past works in NLP have analyzed it as a particle (PART) instead (Akilan and Naganathan, 2012; Ramasamy and Žabokrtský, 2012). The annotation process also surfaced two different types of sentences containing *eṉpatu* which prompted a reanalysis of the function of *eṉpatu*.

Prima facie, the majority of sentences with *eṉpatu* seemed to involve its function as a complementizer, embedding a clause as a noun phrase (NP) that can occur in any NP position (Lehmann, 1993) (see Figure 3). While it has been proposed that *eṉpatu* in such a context can be analyzed as *eṉ-p-atu* (Lehmann, 1993) or even *eṉp-a-atu* (Butt et al., 2020), with the *-atu* suffix in both cases playing a nominalizing role, we find that more work needs to be done before this conclusion can be made and therefore opt to keep the entire word



'It is good that Kumar will not go there'

Figure 3: *Eṉpatu* with complementizer function



'Parvata Kingdom is today's Bhutan'

Figure 4: *Eṉpatu* with copula-like function

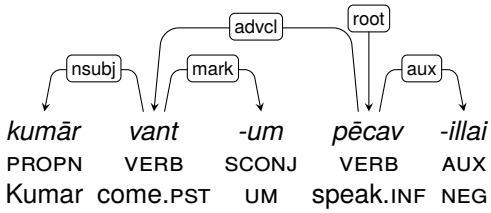
as a single token without splitting it into smaller morphemes. In such a context, we follow the UD guidelines and label *eṉpatu* as SCONJ with a dependency relation of *mark* given its complementizing function.

However, we found that there exists another group of sentences that do not seem to be featuring *eṉpatu* as a complementizer, but rather more like a copula (see Figure 4). As there is no clause with an inflected verb for *eṉpatu* to embed in such a context, it is challenging to analyze *eṉpatu* as a complementizer here. We leave potential reanalysis of this context to future work and opt to label *eṉpatu* as a copula in such contexts, which takes an AUX POS tag and *cop* dependency relation under the UD framework.

This reanalysis of *eṉpatu* as SCONJ and AUX not only clarifies the various functions of *eṉpatu* in different contexts, but is also in line with the recommendations of the UD guidelines to only use the PART label when no other label is possible.

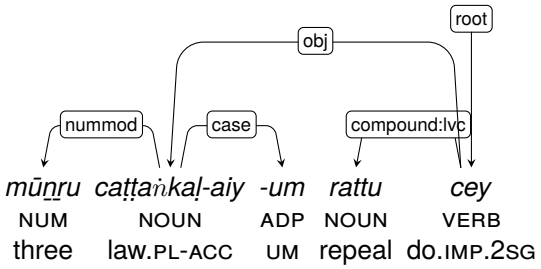
5.2. Clitics

Clitics abound in the Tamil language and serve a plethora of semantic and pragmatic functions (Lehmann, 1993; Schiffman, 1999; Annamalai and Steever, 2019). However, they have not been well studied in previous Tamil treebanking works and are often not treated as separate tokens in their own right. This presents problems in accu-



'Although Kumar came, he did not speak'

Figure 5: *-um* used in a concessive sense



'Repeal all three laws'

Figure 6: *-um* used in an all-inclusive sense

rately determining the dependency relations between words, conflating multiple syntactic and semantic functions in a single token. Therefore, as stated in Section 4.2.1, all clitics in *Aalamaram* were tokenized and rigorous analyses were done to determine their functions.

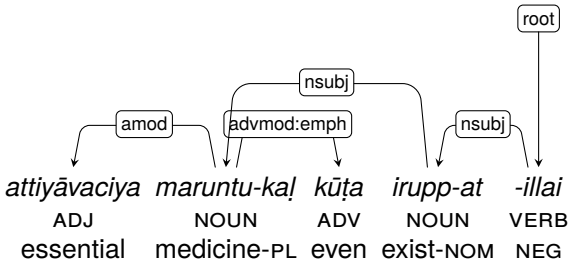
One example would be the particularly polysemous *-um* clitic which can have up to 5 functions based on the examples that we found in *Aalamaram*. In TTB, *-um* can take on a few different dependency relations such as *advmod:emph*, *cc* or *mark*, but the POS tag is always PART. In contrast, in *Aalamaram*, it can take on a POS tag of CCONJ, SCONJ (see Figure 5), ADV, ADP (see Figure 6) or PART depending on the context.

Other clitics that were also tokenized and analyzed include *-ā*, *-āvatu*, *-ām*, *-ē*, *-ō* and *-tān*.

5.3. *illai*

The negative verb *illai* can express negation in both copulative and existential contexts (Lehmann, 1993). It has been suggested that the former should be labeled as AUX with a dependency relation of *cop*, while the latter should be labeled as VERB and should act as the head of the clause (Krishnamurthy and Sarveswaran, 2021). There were two other scenarios in which we found these rules to be insufficient for annotation.

The first scenario involves the use of *illai* as an auxiliary verb when used in the negative form of a main verb (see Figure 5). Such cases were not annotated in MWTT due to the lack of multi-word expansion for words ending in *illai*. A simple rule



'Even essential medicines are unavailable'

Figure 7: *illai* as a main verb

of thumb that we sought to implement was to treat *illai* as an AUX with a dependency relation of *aux* if it is not a standalone token, as the assumption was that the verb it is attached to should be the main verb.

However, the second scenario surfaced while implementing this rule as it was found that there are cases in which *illai* should be interpreted as the main verb when attached to a verb in the future verbal noun form (see Figure 7). While the linguistic arguments supporting this interpretation would require a more in-depth exploration, we tentatively suggest that *illai* be labelled as VERB in such cases.

6. Conclusion

In conclusion, we propose *Aalamaram*, the largest publicly-available dependency treebank for the Tamil language with a size of almost 10,000 sentences manually annotated for POS, morphological features, named entities and dependency relations, with close attention paid to multi-word segmentation. During the process of annotating the treebank, we also discovered various linguistic phenomena in Tamil that prompted reanalysis and adjustment of annotation rules. These include the analysis of clitics, the copula-like function of *eṉpatu*, and the interpretation of *illai* as a main verb or auxiliary. We hope that these discoveries and discussions will enable the field to get closer to a more accurate analysis of Tamil syntax, build more accurate parsers and improve Tamil NLP in general.

Moving forward, there remain certain aspects of the treebank that can be improved. Some possible improvements that can be explored are as follows:

1. More in-depth analyses of suffixes such as *-aana* and *-aaka* and whether multi-word tokenization is warranted for them
2. The use of Enhanced Dependencies[13] to handle linguistic phenomena such as ellipsis

[13] https://universaldependencies.org/u/overview/enhanced-syntax.html

3. Revisions to the Animacy feature to allow intersection of rationality and animacy

4. Further analysis of *illai* and *eṉpatu*

Future work would also include the training of tokenizers, POS taggers, named entity recognizers, morphological parsers and dependency parsers. This could allow us to explore the impact of various annotation decisions on model performance, such as the extensive segmentation of clitics and reanalysis of POS and dependency relations for them.

7. Acknowledgements

This work is supported by the National Research Foundation, Singapore under its AI Singapore Programme. The authors would like to thank the students of Indraprastha Institute of Information Technology, Delhi, India (IIIT-Delhi) and Thiagarajar College of Engineering, Madurai, India (TCE) who were involved in the project for their hard work in annotating the dataset. Additionally, we extend our thanks to Jian Gang Ngui for his invaluable assistance with the linguistic analysis.

8. Bibliographical References

R. Akilan and E. R. Naganathan. 2012. Pos Tagging for Classical Tamil Texts. *International Journal of Business Intelligent*, 1(2):15–17.

E. Annamalai and Sanford B. Steever. 2019. *The Dravidian Languages*, chapter Modern Tamil. Routledge.

Kenneth J. Berry and Paul W. Mielke. 1988. A Generalization of Cohen's Kappa Agreement Measure to Interval Measurement and Multiple Raters. *Educational and Psychological Measurement*, 48:921 – 933.

Emanuele Bugliarello and Naoaki Okazaki. 2020. Enhancing Machine Translation with Dependency-Aware Self-Attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online. Association for Computational Linguistics.

Miriam Butt, S. Rajamathangi, and Kengatharaiyer Sarveswaran. 2020. Mixed Categories in Tamil via Complex Categories. In *Proceedings of the LFG'20 Conference*, pages 68–88, Stanford, CA. CSLI Publications.

Nitish Chandra, Sudhakar Kumawat, and Vinayak Srivastava. 2014. Various Tagsets for Indian Languages and Their Performance in Part of Speech Tagging. In *Proceedings of 5th IRF International Conference*, Chennai, India.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Hiroyuki Deguchi, Akihiro Tamura, and Takashi Ninomiya. 2019. Dependency-Based Self-Attention for Transformer NMT. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 239–246, Varna, Bulgaria. INCOMA Ltd.

Sobha Lalitha Devi, Sindhuja G., Gracy L., Padmapriya N., Gnanapriya A., and Parimala N.H. 2016. AUKBC Tamil Part-of-Speech Corpus (AUKBCTamilPOSCorpus2016v1). Chennai, India. Computational Linguistics Research Group, AU-KBC Research Centre.

V Dhanalakshmi, Anand Kumar, G Shivapratap, KP Soman, and S Rajendran. 2009. Tamil POS Tagging using Linear Programming. *International Journal of Recent Trends in Engineering*, 1(2):166–169.

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.

Jan Hajič, Eduard Bejček, Jaroslava Hlavacova, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpánková. 2020. Prague Dependency Treebank - Consolidated 1.0. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5208–5218, Marseille, France. European Language Resources Association.

Jan Hajič. 1998. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*, pages 106–132.

Yen-Hao Huang, Hsiao-Yen Lan, and Yi-Shin Chen. 2022. Unsupervised Text Summarization of Long Documents using Dependency-

based Noun Phrases and Contextual Order Arrangement. In *Proceedings of the 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)*, pages 15–24, Taipei, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

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*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik 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 the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Elinor Keane. 2004. Tamil. *Journal of the International Phonetic Association*, 34(1):111–116.

P. Krishnamurthy and K Sarveswaran. 2021. Towards Building a Modern Written Tamil Treebank. In *Proceedings of the 20th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2021)*, pages 61–68.

Bhadriraju Krishnamurti. 2003. *The Dravidian Languages*. Cambridge Language Surveys. Cambridge University Press.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Gokul C., Avik Bhattacharyya, Mitesh Khapra, and Pratyush Kumar. 2020. AI4Bharat-IndicNLP Corpus: Monolingual Corpora and Word Embeddings for Indic Languages.

Thomas Lehmann. 1993. *A Grammar of Modern Tamil*, second edition. Pondicherry Institute of Linguistics and Culture publication. Pondicherry Institute of Linguistics and Culture, Pondicherry.

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.

Zuchao Li, Kevin Parnow, and Hai Zhao. 2022. Incorporating rich syntax information in Grammatical Error Correction. *Information Processing Management*, 59(3):102891.

Arnav Mhaske, Harshit Kedia, Sumanth Doddapaneni, Mitesh M. Khapra, Pratyush Kumar, Rudra Murthy, and Anoop Kunchukuttan. 2023. Naamapadam: A Large-Scale Named Entity Annotated Data for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10441–10456, Toronto, Canada. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

R.K. Pattabhi and Sobha Lalitha Devi. 2013. NERIL: Named Entity Recognition for Indian Languages @ FIRE 2013–An Overview. In *Named-Entity Recognition Indian Languages FIRE 2013 Evaluation Track*, FIRE '13, New Delhi, India.

Vijay Sundar Ram R and Sobha Lalitha Devi. 2018. Overview of Verb Phrase Translation in Machine Translation: English to Tamil and Hindi to Tamil. In *Proceedings of the 10th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '18, page 6–10, New York, NY, USA. Association for Computing Machinery.

Loganathan Ramasamy and Zdeněk Žabokrtský. 2012. Prague Dependency Style Treebank for Tamil. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1888–1894, Istanbul, Turkey. European Language Resources Association (ELRA).

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages.

Kengatharaiyer Sarveswaran and Gihan Dias. 2021. Building a Part of Speech tagger for

the Tamil Language. In *2021 International Conference on Asian Language Processing (IALP)*, pages 286–291. IEEE.

Harold F. Schiffman. 1999. *A Reference Grammar of Spoken Tamil*. Reference Grammars. Cambridge University Press.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended Named Entity Hierarchy. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Shashank Siripragada, Jerin Philip, Vinay P. Namboodiri, and C V Jawahar. 2020. A Multilingual Parallel Corpora Collection Effort for Indian Languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3743–3751, Marseille, France. European Language Resources Association.

R Vijayakrishna and Lalitha Devi Sobha. 2008. Domain Focused Named Entity Recognizer for Tamil using Conditional Random Fields. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. OntoNotes: A Large Training Corpus for Enhanced Processing. In Joseph Olive, Caitlin Christianson, and John McCary, editors, *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*, pages 54–63. Springer New York, NY.

Jiacheng Xu and Greg Durrett. 2019. Neural Extractive Text Summarization with Syntactic Compression. 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 3292–3303, Hong Kong, China. Association for Computational Linguistics.

Ryo Yoshida, Taiga Someya, and Yohei Oseki. 2024. Tree-Planted Transformers: Large Language Models with Implicit Syntactic Supervision.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020. SG-Net: Syntax-Guided Machine Reading Comprehension. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Houquan Zhou, Yang Hou, Zhenghua Li, Xuebin Wang, Zhefeng Wang, Xinyu Duan, and Min Zhang. 2023. How Well Do Large Language Models Understand Syntax? An Evaluation by Asking Natural Language Questions.

Author Index