



LREC 2026

**The 7th Workshop on Open-Source Arabic Corpora and
Processing Tools (OSACT7) with 5 Shared Tasks**

Workshop Proceedings

Editors
Hend Al-Khalifa
Mo El-Haj
Saad Ezzini

May 11, 2026

Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks

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

ISBN 978-2-493814-52-4
EAN 9782493814524

Preface

Welcome to the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT2026), co-located with LREC 2026. This workshop brings together researchers and practitioners working on Arabic Natural Language Processing (NLP), with a focus on open-source resources, tools, and systems that advance the state of the art in Arabic language technologies.

Arabic NLP continues to present unique and exciting challenges, driven by the rich morphological complexity of the language, the diversity of its dialects, and the growing demand for high-quality tools and resources across a wide range of applications. At the same time, the rapid advancement of Large Language Models (LLMs) has opened new research directions, raising important questions about their effectiveness, limitations, and adaptability for Arabic.

For this edition, we received 35 submissions covering a broad range of topics. After a thorough double-blind review process, we accepted 16 papers, yielding an acceptance rate of approximately 46%. The accepted papers are organized into two main thematic sessions: the first focusing on Sentiment Analysis, Information Retrieval, and Language Resources, and the second on Machine Translation, Dialects, and LLM Applications.

In addition, the workshop hosted five shared tasks spanning Arabic politeness detection, Islamic inheritance reasoning, speech dictation with automatic diacritization, humor generation, and sentiment analysis. These shared tasks attracted a total of 33 system paper submissions, reflecting the strong community engagement and growing interest in building shared benchmarks and evaluation frameworks for Arabic NLP.

We thank the members of the program committee for their thorough and thoughtful reviews. We also thank the organizers of LREC 2026 for their continued support. Most importantly, we thank all the authors and participants whose contributions and enthusiasm made this workshop a vibrant and valuable venue for the Arabic NLP community.

Hend Al-Khalifa
Mo El-Haj
Saad Ezzini

Organizing Committee

Hend Al-Khalifa, King Saud University (KSU), Saudi Arabia
Mo El-Haj, Lancaster University, United Kingdom; VinUniversity, Vietnam
Saad Ezzini, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia

Program Committee

Mohammed Alliheedi, Al-Baha University, Saudi Arabia
Mamoun Abu Helou, Al-Istiqlal University, Palestine
Maria Bounnit, Cadi Ayyad University, Morocco
Hani M. Iwida, Al-Istiqlal University, Palestine
Noorhan Abbas, University of Leeds, United Kingdom
Sultan Alrowili, IBM Research, Saudi Arabia
Abdulaziz Alhamadani, Florida Polytechnic University, United States
Amal Haddad Haddad, University of Granada, Spain
Dima Taji, Charles University, Czech Republic
Hamada Nayel, Prince Sattam Bin Abdulaziz University, Saudi Arabia
Salmane Chafik, Mohammed VI Polytechnic University, Morocco
Abdessalam Boucekif, Hamad Bin Khalifa University, Qatar
Sharefah Al-Ghamdi, King Saud University (KSU), Saudi Arabia
Akram Mohammed Ahmed Al-Rumaim, Independent Researcher, Netherlands
Ashraf Elnagar, University of Sharjah, United Arab Emirates
Ameera Almasoud, King Saud University (KSU), Saudi Arabia
Imed Zitouni, Meta, United States
Mohammad Abuoudeh, Al-Hussein Bin Talal University, Jordan
Wajdi Zaghouani, Northwestern University in Qatar, Qatar
Rana Malhas, Qatar University, Qatar
Ehsan Lotfi, University of Antwerp, Belgium
Abdelkader El Mahdaoui, Mohammed VI Polytechnic University, Morocco
Khalid Choukri, ELRA/ELDA, France
Khalid Al-Khatib, University of Groningen, Netherlands
Amany Fashwan, Alexandria University, Egypt
Salima Harrat, ENS Bouzaréah, Algeria
Manal Albahlal, King Saud University (KSU), Saudi Arabia
Wafa Aissa, UCLouvain, Belgium
Bassam Haddad, University of Petra, Jordan
Noof Alfear, King Saud University (KSU), Saudi Arabia
Amr Keleg, Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), United Arab Emirates
Sara Nabhani, University of Groningen, Netherlands
Ali Al-Laith, University of Copenhagen, Denmark
Fethi Bougares, Avignon University, France
Amal Htair, Aston University, United Kingdom
Nourah Alangari, King Saud University (KSU), Saudi Arabia
Nadia Ghezaiel, University of Ha'il, Saudi Arabia
Khlood Al Jallad, Syrian Society for Startups and Research, Syria
Paul Rayson, Lancaster University, United Kingdom

Hamza Alami, Sidi Mohamed Ben Abdellah University (USMBA), Morocco
Abdulhamid Abubakar, Nasarawa State University (NSU), Nigeria
Kamel Gaanoun, National Institute of Statistics and Applied Economics (INSEA), Morocco
Mariam M. Biltawi, Al Hussein Technical University, Jordan
Abdullah I. Alharbi, King Salman Global Academy for Arabic Language, Saudi Arabia
Marwah Alian, World Islamic Sciences and Education University, Jordan
Sarah Alnefaie, King Abdulaziz University, Saudi Arabia
Hatim Derrouz, Ibn Tofail University, Morocco
Nada Ghneim, Damascus University, Syria
Taha Zerrouki, University of Bouira, Algeria
Fouzi Takelait, University of North Dakota, United States
Serry Sibae, Prince Sultan University, Saudi Arabia
Yassine Saoudi, Al-Ahliyya Amman University, Jordan
Hadeel Saadany, Birmingham City University, United Kingdom
Salima Mdhaffar, Avignon University, France
Nouran Khallaf, University of Leeds, United Kingdom
Amir Hazem, The University of Tokyo, Japan
Osama Hamed, Palestine Technical University, Palestine

Table of Contents

<i>Hidden Sentiments: The Impact of Low-level Adversarial Perturbations on Arabic Sentiment Analysis Services</i>	
Abdelrahman Hamada Hefny Abdelkader	1
<i>LLM-Based Financial Sentiment Analysis in Arabic: Evidence from Saudi Markets</i>	
Mona H. Albaqawi, Eman M. Albalkhi, Joud A. Albaiti and Enrico Lopedoto	14
<i>Does Translation Preserve Sentiment? An Analysis of Arabic-English Cross-Lingual Classification</i>	
Nour Aldin Al Mubarak and Noura Al Moubayed	25
<i>When Bigger Isn't Better: Evaluating LLMs for Arabic Sentiment Analysis</i>	
Mohamed Ibrahim, Abdullah Makki, Youssef Barakat, Nour Samy and Sarah AlHumoud	35
<i>GATE-Reranker: A Strong Arabic Cross-Encoder for Document Reranking</i>	
Omer Nacar, Omar Elshehy, Mohamed Zaytoon and Khlood Al Jallad	40
<i>How Foundation Models Behave for Arabic Image Captioning?</i>	
Khaoula Dahimi, Amel Belabbaci, Hadda Cherroun and Abdelhamid Haouhat	49
<i>AlignAR: Generative Sentence Alignment for Arabic–English Parallel Corpora of Legal and Literary Texts</i>	
Baorong Huang and Ali Asiri	59
<i>Helpful or Harmful? The Dual Role of Linguistic Features in LLM-Based Dialectal Machine Translation</i>	
Abdelhalim Hafedh Dahou and Mohamed Amine Cheragui	66
<i>ASCAT: Arabic Scientific Benchmark for Advanced Translation Evaluation</i>	
Serry Sibae, Khlood Al Jallad, Zineb Yousfi, Israa Elhosiny, Yousra Yousra El-Ghawi, Batool Balah and Omer Nacar	76
<i>SHEINfer: Implicit Product Category Inference from Arabic E-commerce Reviews</i>	
Hend Al-Khalifa	81
<i>NAJD-MT: High-Fidelity Saudi Najdi–English Training Data for Bidirectional Neural Machine Translation</i>	
Nour Qandos, Samar Essa Ahmed, Omer Nacar, ahmad alrabghi, Rahaf Saeed Al Hallay, Aya Hamod and Shaden Alsuhaim	88
<i>Parsing Arabic Dialects Revisited: New Benchmarks, Models, and Insights</i>	
Ahmed Farouk Zakaria Elshabrawy, Go Inoue, Muhammed AbuOdeh and Nizar Habash	94
<i>On LLM Prompting Techniques for Arabic Language Arithmetic Reasoning</i>	
Reem Alenezi and Ayed Atallah Salman	106
<i>DIA2 - a Comprehensive and Diverse Diacritized Arabic Corpus for NLP Research</i>	
Fatima Dekmak, Shady Elbassuoni, Khaled Shaban, Hazem Hajj, Wassim El-Hajj, Yasmine Abu Adla and Buthaina Alabrash	115

<i>CV-18 NER: Augmented Common Voice for Named Entity Recognition from Arabic Speech</i> youssef saidi, Haroun Elleuch and Fethi Bougares	131
<i>ARHAHA 2026: The Shared Task on Arabic Humor Automatic Generation</i> Ameera Masoud Almasoud, Hend Al-Khalifa, Reem Fahad Alqifari, Nourah Alangari and Manal M. Albahlal	141
<i>The AdabEval 2026 Shared Task on Arabic Politeness Detection</i> Reem Fahad Alqifari, Hend Al-Khalifa, Nadia Ghezaiel, Maria Bounnit, Hend Hamed Al- hazmi, Ameera Masoud Almasoud, Sharefah Ahmed Al-Ghamdi and Noof Abdullah Alfear	151
<i>GHAD NLP at AdabEval2026: Transformer-Based Approach for Arabic Politeness and Prag- matic Category Classification</i> Ghada Alfattni and ghader kurdi	157
<i>MOSKA-NLP at AdabEval 2026: Feature-Enriched Ensembling for Arabic Politeness Detection</i> Nina A. Andriyanova-Almaamary	165
<i>SHU at AdabEval 2026: Category-Aware Fine-Tuning of MARBERT for Arabic Politeness and Pragmatic Function Classification</i> Alla Zwawi and Stephen Wu	174
<i>Comparative Study of Machine Learning and Transformer-Based Approaches for Arabic Polite- ness Detection at AdabEval 2026</i> mariem ben arbia, Ghada Ben Amor and Omar Trigui	179
<i>AdabEval 2026 Task B : Multi-Label Classification of Arabic Politeness Criteria in Social Media Media</i> Rand Abdullah Alturki	185
<i>QIAS 2026: Overview of the Shared Task on Islamic Inheritance Reasoning</i> Abdessalam Boucekif, Somaya Eltanbouly, Shahd Gaben, Mohammed Ghaly, Samer Rashwani, MOHAMED Emad and Heba Sbahi	191
<i>QU-NLP at QIAS 2026: Multi-Stage QLoRA Fine-Tuning for Arabic Islamic Inheritance Reason- ing</i> Mohammad ALSmadi	199
<i>PSL at QIAS 2026: Which Models Perform Better in Arabic Inheritance Reasoning?</i> amine Mohammed and Chahinez Boucekif	204
<i>AGS-KSU at QIAS 2026: A Comparative Study of Prompting and LLM Approaches for Structured Islamic Inheritance Reasoning</i> Hicham Ghazi Sidaoui	209
<i>Silah at QIAS 2026: Fine-Tuning vs. Retrieval-Augmented Generation for Islamic Inheritance Reasoning</i> Ghader Kurdi, Hanan Justanieah and Hala Justanieah	213

KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization

Asma Ali Al Wazrah, Waad Alshammari, Rawan Almatham, Raghad Al-rasheed, Afrah Abdulaziz Altamimi, Rufael Marew, Sawsan Alqahtani, Hanan Aldarmaki, Abdullah I. Alharbi, Abdulrahman Saeed Alshehri, Mohamed Assar, Amal Almazrui and Abdulrahman Alosaimy
220

Thaka at KSAA-2026 Task 2: Regularized Fine-Tuning for Arabic Speech Diacritization

Meshal Abdullah Alamr, Hassan Rshed Alqaeri and Abdullah Aldahlawi 225

TantaArabNLP at KSAA-2026 Task 2: Adapting CATT-Whisper for Arabic Speech Dictation with Automatic Diacritization

Nada Adel Esmaeil, Reda M. Elbasiony and Mohamed T. Faheem 229

Fine-Tashkeel at KSAA-2026: A Comprehensive Evaluation of Seq2Seq and Multimodal Approaches for Automatic Diacritization of Arabic Speech Dictation

Hassan Barmandah, Fatimah Emad Eldin, Omer Nacar and Wareef Alzubaidi 234

Eraserhead at OSACT7 Shared Task: ASR Consistency Filtering and Speaker-Adaptive Post-Processing for Arabic Speech Diacritization

Muhammad Abu Horaira and Nahian Chowdhury 247

Abjad AI at KSAA-2026 Shared Task 2: Grouped Speech Conditioning for Arabic Diacritization

Naif Saad Alharthi, Ahmad Ghannam, Faris Alasmay, Kholood Al Tabash, Shouq Sadah and Lahouari Ghouti 252

AraSentEval 2026: A Shared Task on Sentiment Analysis and Swapping in Arabic

Saad Ezzini, Shadi Abudalfa, Maram I. Alharbi, Salmane Chafik, Hamzah Luqman, Mo El-Haj, Paul Rayson and Reem Alotaibi 256

TTLab at AraSentEval: SARF Sentiment Analysis via Root-based Fusion for Multi-Dialectal Arabic

Ali Abusaleh, Bhuvanesh Verma and Alexander Mehler 262

A Comparative Study of Arabic Sentiment Swap Models for AraSentEval 2026

Yumna Hamdy, Mohab ElDamhougy, Yomna Eid and Ensaf Hussein 269

L3IA at AraSentEval 2026 Subtask 2: LLM-Based Multi-Step Pipeline for Arabic Sentiment Swap

Abdessamad Benlahbib, Hamza Alami, Mohamed M'haouach and Kaouthar Elyoussoufi
274

CasbAI at AraSentEval 2026: Robust Dialectal Arabic Sentiment Classification via Multi-Seed Ensembling and Data Augmentation.

chaima abdelaziz, KahinaHouda Saadaoui, Faiza BELBACHIR and Lynda Said Lhadj 278

BDSI at AraSentEval Shared Task : A Multi-Transformer Contrastive Learning for Arabic Dialect Sentiment Analysis

Mohamed M'haouach, Kaouthar Elyoussoufi, Abdessamad Benlahbib and Hamza Alami
284

Codezone Research Group at AraSentEval Shared Task: Arabic Sentiment Swap beyond Negation Prepending, Benchmarking Multilingual T5 against Large Language Models on the MA'AKS Corpus

Abdulkadir Shehu Bichi and Sarah Yassine 288

L3IA-Subtask 1 at AraSentEval Shared Task: Multi-Dialect Arabic Sentiment Classification via a Transformer-Based Approach

Mohamed M'haouach, Kaouthar Elyoussoufi, Hamza Alami and Abdessamad Benlahbib
292

University of Tripoli at AraSentEval: Fine-Tuning MARBERTv2 and CAMELBERT for Multi-Dialect Arabic Sentiment Analysis

Abdusalam F. Ahmad Nwesri, Amani Bahlul Sharif and Sarah Farag S. Hmeid 296

Workshop Program

09:00–09:05 Opening Remarks

Session 1: Sentiment Analysis, Information Retrieval & Language Resources

09:05–10:25

09:05–09:15	Hidden Sentiments: The Impact of Low-level Adversarial Perturbations on Arabic Sentiment Analysis Services
09:15–09:25	LLM-Based Financial Sentiment Analysis in Arabic: Evidence from Saudi Markets
09:25–09:35	When Bigger Isn't Better: Evaluating LLMs for Arabic Sentiment Analysis
09:35–09:45	Does Translation Preserve Sentiment? An Analysis of Arabic-English Cross-Lingual Classification
09:45–09:55	GATE-Reranker: A Strong Arabic Cross-Encoder for Document Reranking
09:55–10:05	How Foundation Models Behave for Arabic Image Captioning?
10:05–10:15	AlignAR: Generative Sentence Alignment for Arabic–English Parallel Corpora of Legal and Literary Texts
10:15–10:25	Helpful or Harmful? The Dual Role of Linguistic Features in LLM-Based Dialectal Machine Translation

10:30–11:00 Coffee Break

Session 2: Machine Translation, Dialects & LLM Applications

11:00–12:15

11:00–11:12	ASCAT: Arabic Scientific Benchmark for Advanced Translation Evaluation
11:12–11:24	NAJD-MT: High-Fidelity Saudi Najdi–English Training Data for Bidirectional Neural Machine Translation
11:24–11:36	On LLM Prompting Techniques for Arabic Language Arithmetic Reasoning
11:36–11:48	Parsing Arabic Dialects Revisited: New Benchmarks, Models, and Insights
11:48–12:00	DIA2 – a Comprehensive and Diverse Diacritized Arabic Corpus for NLP Research
12:00–12:12	CV-18 NER: Augmented Common Voice for Named Entity Recognition from Arabic Speech

Workshop Program (Cont.)

Session 3: Arabic NLP Shared Tasks

12:15–12:55

12:15–12:23 **ARHAHA 2026: The Shared Task on Arabic Humor Automatic Generation**

12:23–12:35 **The AdabEval 2026 Shared Task on Arabic Politeness Detection**
• GHAD NLP at AdabEval 2026 (lightning talk)

12:35–12:43 **QIAS 2026: Overview of the Shared Task on Islamic Inheritance Reasoning**
• QU-NLP at QIAS 2026 (lightning talk)

12:43–12:50 **KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization**
• Thaka at KSAA-2026 (lightning talk)

12:50–12:55 **AraSentEval 2026: A Shared Task on Sentiment Analysis and Swapping in Arabic**
• TTLab at AraSentEval 2026 (lightning talk)

12:55–13:00 Closing Remarks

Hidden Sentiments: The Impact of Low-Level Adversarial Perturbations on Arabic Sentiment Analysis Services

Abdelrahman H. Hefny

Carleton University
Ottawa, Canada
abdelrahmanhefny@gmail.carleton.ca

Abstract

Sentiment analysis is one of the most popular applications of supervised machine learning for natural language processing. A common approach for obtaining a dataset to train sentiment analysis models is to extract user posts and comments from social media and other online platforms. However, this content is subject to various types of perturbations that go beyond the target of common preprocessing techniques and may impact the models' performance. In this paper, a set of six popular corpora used in Arabic sentiment analysis research is analyzed to identify common patterns of character-level perturbations. The samples of three selected corpora were then used to test the performance of the online sentiment analysis services offered by three public cloud providers. This test is done using a clean version of each dataset and four other versions, each perturbed using a different technique. Empirical results indicate that no single sentiment analysis service is superior to others in all cases, and all three services are vulnerable to low-level adversarial attacks which may cause up to a 51% relative drop in macro average F1 score, while maintaining readability.

Keywords: Arabic NLP, Sentiment Analysis, Adversarial Attacks, Cloud Services

1. Introduction

Since the emergence of social media platforms, the amount of user-generated content on the web has been increasing dramatically. This content in the form of text, images, audio, video, and user interactions provides a great source of insights for businesses and decision makers, allowing them to better understand their audiences and make informed decisions. However, with the huge amount of data generated every second, it's not feasible anymore to monitor and analyze this content manually, which makes the use of automated Machine Learning (ML) and data mining techniques a necessity. These techniques can be utilized in many areas. One such area is sentiment analysis of textual content. Sentiment Analysis (SA) is the use of Natural Language Processing (NLP) techniques to analyze sentiments in a given input text, and classify the text as either positive, negative, or neutral. Applications of sentiment analysis include, for example, aggregating user feedback on new products or services, polling public opinions toward a specific subject, predicting stock prices, or forecasting election outcomes (Medhat et al., 2014).

Currently, the state of the art in sentiment analysis uses ML by training a model on a large dataset containing examples of all possible classes (Jim et al., 2024). The better the quality of training data, the better the performance of the trained model on unseen examples. Researchers working on sentiment analysis are concerned with the choice of preprocessing steps applied to input data, which often involves cleansing and normalization to improve the model's performance. For Arabic, this might

include, for example, removing punctuation marks, diacritics, and stop words, stemming words, or normalizing each letter to a single form.

However, as sentiment analysis is often applied to textual content obtained from social media platforms and online stores, inputs might be subject to varying levels of perturbation, as in Figure 1, that alter the text in ways that go beyond the target of simple normalization techniques, resulting in degraded performance of NLP and text mining algorithms including sentiment analysis. Such perturbations require a more advanced level of preprocessing to mitigate their effects which is often neglected in many production ML systems. Furthermore, since research on Arabic sentiment analysis utilizes sample datasets collected from the same sources, these samples are also subject to the same perturbations, and, depending on the dataset size, they might affect the model's performance on the test sample.

These perturbations could be intentional to evade content moderation, work around character limits of some platforms such as X (formerly Twitter), or just create an artistic style, but they could also be accidental. They could be perceptible to humans, such as spelling mistakes, omitted spaces between words, redundant diacritics, or words obfuscated with punctuation symbols. Yet, they could also be imperceptible, such as invisible control characters or characters replaced with others that have similar shapes (homoglyphs).

Text perturbations intended to degrade the performance of NLP models are a form of adversarial attacks against ML systems, which are a major

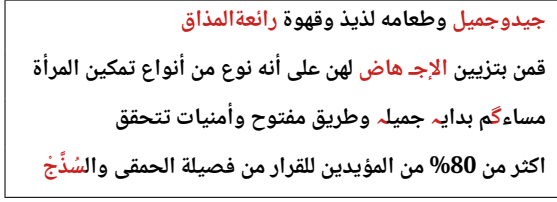


Figure 1: Examples of perturbations in SA corpora.

concern for ML security specialists and an active research area (Zhou et al., 2022). Several recent studies investigated adversarial attacks on computer vision and NLP models (Ren et al., 2020; Xu et al., 2020; Goyal et al., 2023; Costa et al., 2024). However, most adversarial NLP research has focused on English and Latin script languages, while relatively few studies were concerned with Arabic. Nevertheless, the differences between Arabic and Latin script languages make some techniques that are intended for English inapplicable to Arabic and vice versa.

The goal of this research is to investigate the effects of adversarial text perturbations applied at the low (character) level on the performance of SA models. Instead of testing open-source NLP models which are open to retraining, finetuning, and customization, the focus of this work is on production-ready SA services offered by major cloud service providers as part of their Artificial Intelligence (AI) services catalog. These services are often offered as a black box with little or no customizability. However, they provide software developers with easy to access Application Programming Interfaces (APIs) backed by pretrained ML models to utilize in their software applications without having to build or maintain such models by themselves. A client application sends a secure request over the Internet to an API endpoint with one or more attached documents, and the service processes the request and replies with the predicted sentiment of each document. Some services also can predict the sentiment of each sentence in the document, and link sentiments to named entities in the text.

To achieve this goal, a set of corpora commonly used in Arabic SA research is first analyzed to identify common patterns of low-level text perturbations that might affect NLP models performance. Then a subset of these corpora is sampled and used for evaluating the performance of Arabic SA services on three popular cloud platforms. This evaluation is done once on the clean datasets, and once on four versions of the datasets with different types of perturbations.

This paper makes the following contributions:

1. A list of identified low-level text perturbations found in a sample of Arabic SA corpora.

2. A comparison of Arabic sentiment analysis services offered by three popular cloud providers: Amazon Web Services, Microsoft Azure, and Google Cloud Platform, on three different sample corpora covering dialectal and modern standard Arabic.

3. An evaluation of the selected services on different versions of the test samples perturbed using four techniques of low-level adversarial attacks.

This research uses publicly available datasets, and all the code used for analysis and experimental evaluation is also available through the project's code repository.¹ The remainder of this paper is organized as follows. Section 2 provides a brief review of related work. Section 3 describes the research methodology, including the selected cloud services, datasets, and experiment design. Section 4 discusses the main research findings. Finally, Section 5 highlights the research conclusions.

2. Related Work

The following subsections provide a brief review of the literature on related work under three main topics: Arabic sentiment analysis, adversarial attacks on NLP, and evaluation of cloud-based NLP services, respectively.

2.1. Arabic Sentiment Analysis

Sentiment analysis is one of the most studied applications of NLP and text classification. Traditionally, there have been three main approaches for sentiment analysis: a lexicon-based approach, a supervised ML approach, or a hybrid approach. (Wankhade et al., 2022; Mao et al., 2024) However, with the availability of large training datasets and the recent advances in NLP, Deep Learning (DL), and language models, using ML/DL models has become the state-of-the-art in sentiment analysis for many languages, including Arabic (Jim et al., 2024). Research on Arabic SA has covered many domains, such as movies, hotels, books, mobile applications, vaccines, and political views. As the number of publications on Arabic SA is relatively large, covering them in detail is outside the scope of this paper. However, several recent surveys are available for reference (Oueslati et al., 2020; Obiedat et al., 2021; Matrane et al., 2023).

¹<https://github.com/abdelrahman0101/HiddenSentiments>

2.2. Adversarial Attacks on NLP Models

Adversarial attacks against ML models are perturbations applied to the models' inputs in order to hinder their performance. (Xu et al., 2020) These attacks are categorized based on knowledge of the target system into white-box, where the attacker has full knowledge of the model's architecture and access to internal system parameters, and black-box, where they can only get the system's output in response to a given input (Ren et al., 2020). In NLP, they can be further categorized according to the level of perturbation as sentence, word, or character level. These perturbations introduce minor insertions, deletions, or substitutions such that humans are able to perceive the original information, while ML models fail to do so (Alshemali and Kalita, 2020; Qiu et al., 2022; Goyal et al., 2023). Such attacks are known to cause performance degradation in sentiment analysis (Radman and Duwairi, 2025), machine translation (Zhang et al., 2021; Sadrizadeh et al., 2023), offensive language detection (Cooper et al., 2023; Abdellaoui et al., 2024; Berezin et al., 2025), plagiarism detection (Alvi et al., 2017; Creo and Pudasaini, 2025), spam detection (Alajmi et al., 2025), and other tasks (Xie et al., 2022).

Studies on adversarial attacks on NLP models were mostly focused on English and Latin script languages, while a relatively small portion targeted Arabic. For Latin script languages, examples of possible perturbations include intentional typos (Gan et al., 2024), insertion of punctuations (Formento et al., 2023), homo-glyph substitutions (Eger et al., 2019; Cooper et al., 2023), and insertion of invisible formatting characters (Boucher et al., 2022). As for Arabic, examples include misspellings that mimic the mistakes of non-native speakers (Alshemali and Kalita, 2021), substituting dotted characters with their dotless counterparts (Al-Shaibani and Ahmad, 2023), insertion of dots or spaces (Abdellaoui et al., 2024), or diacritical manipulations (Salman et al., 2024; Alshemali, 2025a). Some studies also covered word and sentence level attacks (Alshalan and Rekabdar, 2023; Salman et al., 2024; Alshemali, 2025b).

2.3. Evaluation of cloud-based NLP services

Cloud-based AI services have been the target of several recent studies aiming to compare and evaluate their performance on specific benchmarks. Examples are content moderation services used for filtering undesirable content (AIDahoul et al., 2023; Zheng et al., 2024), and natural language understanding (NLU) services used for building chatbots and conversational agents (Braun et al., 2017; Liu et al., 2021). Sentiment analysis ser-

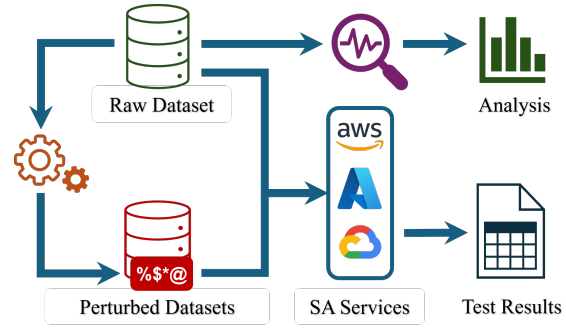


Figure 2: Research process

vices were also the subject of some recent publications (White and Rege, 2020; Ermakova et al., 2023; Ivan et al., 2024; Pawlik, 2025). However, they were also mostly focused on English.

3. Research Methodology

As depicted in Figure 2, this research involves three parts. The first is an exploratory analysis of six Arabic SA corpora, for the purpose of identifying any existing text perturbations. The second part provides an experimental evaluation of three cloud-based services on samples extracted from three selected corpora containing examples in dialectical and modern standard Arabic. In the final part, the same tests are repeated, but this time on versions of the samples perturbed using four different techniques, and the results are compared to the baseline obtained from the previous experiment.

3.1. Exploratory Analysis of Existing Corpora

The author surveyed existing research on Arabic SA published in the last 10 years and selected a set of six well-cited corpora in the academic literature with varying domains, sources, and dialects, which are listed in Table 1. These corpora were mainly obtained from social media platforms and each record was annotated to indicate its sentiment using either a nominal label or a numeric rating. For nominal labels, at least two labels covering positive and negative sentiments are used. Some labeling schemes also add a neutral/objective label for texts not expressing any sentiments and a mixed label for texts with both sentiments.

Based on the author's knowledge and after reviewing a sample of these corpora, a small Python script was implemented to detect potential signs of text perturbation. The script searches through a given dataset and identifies any instances of the

Corpus	Source	Size	Label Scheme	Main Dialect
ASTD (Nabil et al., 2015)	Twitter	10K	4 classes	Egyptian
Elsahar2015 (ElSahar and El-Beltagy, 2015)	Multiple Websites	34K	3 classes	Multiple
AraSenTi-Tweet (Al-Twairish et al., 2017)	Twitter	18K	4 classes	Saudi
OCLAR (Al Omari et al., 2019)	Google Maps, Zomato	4K	5-points rating	Lebanese
NADI 2022 (Abdul-Mageed et al., 2022)	Twitter	5K	3 classes	Multiple
Mawqif (Alturayef et al., 2022)	Twitter	4K	3 classes	Multiple

Table 1: Selected Arabic sentiment analysis corpora

following potential perturbation and tags them for further review.

Non-standard characters. Including any Unicode characters not available on standard Arabic, English, or French keyboards, which are commonly used by Arab users, while excluding Unicode blocks for emoticons, pictographs, and common symbols.

Overly long words. This is to detect posts containing two or more consecutive words concatenated together. The longest known word in classic Arabic lexicons is 15 letters long, but for the purposes of this research a limit of 10 letters was sufficient to identify potential instances.

Arabic words split using spaces, punctuation, emoticons, or Latin letters. This is a common obfuscation technique that usually relies on Kashida (elongation) characters to maintain the readability of the word, which could be mostly detected using regular expressions.

No dotted letters. Most Arabic letters are written with dots or hamza above or below. Another obfuscation technique used in social media is to replace Arabic letters with their undotted forms.

The posts identified through this search were then manually reviewed to remove false positives and identify existing patterns of text perturbation.

3.2. Baseline performance

For practical reasons, only three of the surveyed corpora were selected for the experimental evaluation of cloud-based Arabic SA services. These are ASTD, AraSenTi-Tweet, and Mawqif. The three corpora contain samples in Modern Standard Arabic (MSA) in addition to one or more Arabic dialects. However, to work within the free trial limits of the selected services, and to avoid throttling the APIs, only a random sample containing 100 instances of each class was selected for testing.

The selected cloud services are Amazon Comprehend from Amazon Web Services (AWS), Azure AI Language from Microsoft Azure, and

Cloud Natural Language from Google Cloud Platform (GCP). These platforms are the most popular public cloud providers, respectively, offering a wide range of services. While Azure and AWS APIs use a 4-class labeling scheme, GCP’s uses a numeric score in the continuous range $[-1, 1]$. Thus, to facilitate comparisons, this range is split evenly into 3 subranges of equal length. Scores between -0.3 and 0.3 inclusive are considered neutral, while lower scores are considered negative and higher scores positive. In addition, for tests involving GCP or the Mawqif corpus, the mixed and neutral classes are considered equivalent. Other sentiment analysis tasks such as stance detection are not covered by this research due limited Arabic support of the tested services at the time of this writing.

3.3. Performance on Adversarial Inputs

After setting a baseline for each service on each dataset. Four different versions of the datasets are created by applying the following low-level perturbations:

- **Homoglyphs:** Arabic letters are replaced with Unicode homoglyphs from other Arabic script languages. To maintain the same level of readability, only homoglyphs without extra dots or sub-glyphs are used.
- **Split words:** Every word longer than 3 letters is split using spaces or punctuation symbols.
- **Positional variants:** Every Arabic letter is replaced with its hard-coded positional variant according to its position in the word: initial, middle, final, or isolated.
- **Diacritics:** Arabic diacritics denoting short vowels are added after every consonant using a simple rule-based algorithm based on word syllables.

This experiment is conducted to test the hypothesis that online SA services are vulnerable to adversarial character-level perturbations, and these perturbations can significantly reduce the services' performance, particularly by causing subjective (positive or negative) documents to be classified as objective (neutral).

4. Results and Discussions

The following subsections list and discuss the findings of each of the three research parts.

4.1. Text Perturbations in Existing Corpora

After examining the sample corpora, several cases of low-level text perturbations were identified. Example posts are listed in Table 6 in the appendix. These perturbations can be categorized into the following seven patterns. Nonetheless, it's possible that other types of perturbations exist in the selected corpora, but they were not identified by the search script or during manual review. Also, contrary to the initial expectations, no posts written in undotted letters were found in the selected corpora.

4.1.1. Positional variants of Arabic letters

The most frequently used non-standard characters in the examined corpora were hard-coded positional/contextual variants of Arabic letters from the Unicode block Arabic Presentation Forms B. These Unicode codepoints are provided for backward compatibility with legacy systems where a different codepoint is used for every form of an Arabic letter depending on its position in the word: initial, middle, final, or isolated. Eight of the top ten non-standard characters in the corpora fall under this category, with an average frequency of 117 posts. Despite being completely different on the binary level, it's easy to map these characters to their canonical forms using normalization tools readily available in standard libraries of many programming languages.²

4.1.2. Homoglyphs from other alphabets

Homoglyphs from Arabic script languages such as Persian, or Urdu, were also used in many cases to replace standard Arabic letters. In most cases, they were used for creating a decorative style, but in some cases, they were also used to represent specific sounds used in dialectical Arabic. Examples are Heh Goal (هـ) which was used in 96 posts,

Gaf (گ) in 94 posts, and Peh (پ) in 34 posts. Homoglyphs from other writing systems were also found. Examples are Hebrew Vav (ו) and Greek Iota (ι) used to mimic Arabic Alef. Mitigating this kind of perturbation is easier when the text language is known in advance. In such a case, it's a matter of character mapping. However, this mapping is not always one to one as the same homoglyph can be used for different Arabic letters in different positions.

4.1.3. English sounds and Arabizi

Instead of using homoglyphs from other languages, an Arabic letter can be replaced with an English/Latin letter that has a similar sound, or with a digit using the Arabizi mapping. The resulting word contains Arabic letters intermixed with digits or Latin letters. This is different from writing an entire word in English letters or Arabizi, which is outside the scope of this paper. This pattern was found in very few posts where words were perturbed mostly to evade content moderation.

4.1.4. Concatenated words

Although the search was focused on overly long words with a minimum length of 11 letters, it was able to identify several instances of concatenated Arabic words. In some cases, this was likely the result of typos, but in many cases, the first word ended with a terminal letter that does not connect to its following neighbors. This suggests that spaces were deliberately omitted either as a typing habit or to confine with the length limitation imposed by the platform. Mitigating this kind of perturbation requires the use of a spellchecker that supports dialectical Arabic, but some terminal letters that only occur at the end of words such as Taa Marbuta (ة) can be used to determine where a space should be inserted.

4.1.5. Split words

Instead of merging consecutive words, some words were split into two or more parts using extra spaces or punctuation marks, along with Kashida characters to maintain readability. In most detected cases, it separated a word prefix from its stem, which might in fact facilitate keyword matching, but also might hinder morphological or syntactical parsing. Nonetheless, in some cases, it was also used to split a word stem, which is commonly used with words related to sensitive topics that might be flagged by content moderation. Regular expressions can be used to detect many potential cases of this perturbation, but a dialectical spell checker is still needed to mitigate them.

²Examples are the *unicodedata* module for Python, and the *java.text.Normalizer* class for Java.

4.1.6. Invisible formatting characters

Formatting and control characters are non-printing characters that have no visible glyphs but can significantly affect how text is displayed and formatted. They can be used to delete characters, reverse the display order, or split a word. Invisible characters found in the corpora include Zero-Width Joiner (43 posts), Right-to-Left Mark (14 posts), and Zero-Width Non-Joiner (11 posts). They were mainly used to enhance the readability of surrounding text, especially when homoglyphs or positional variants are used, or when the text contains emoticons or non-Arabic left-to-right words. However, any adversarial perturbations involving these characters can be easily mitigated using a rule-based approach.

4.1.7. Diacritics

Diacritical marks are an essential part of Arabic script. A subset of these marks used for denoting short vowels in Arabic are available on standard keyboard schemes. These were used in over 3200 posts in the sample corpora. However, other diacritics were also used for decorating text, including ones from other writing systems and the Combining Diacritical Marks block. For normalizing inputs to Arabic NLP models, it's mostly considered safe to simply remove diacritics from Arabic text.

4.2. Baseline Performance

The first experiment was conducted on samples extracted from the three selected corpora. Evaluation includes two sentiment analysis tasks with different numbers of target classes. The first task uses a four-classes scheme to classify an input as Positive, Negative, Neutral, or Mixed. This, however, is only possible with the platforms and datasets that support this scheme. The second task uses a three-classes scheme, by treating the Neutral and Mixed classes as equivalent. Before taking the samples, cleansing and normalization steps were performed to ensure they are mostly free from identified perturbations. First, all posts were normalized to Unicode's Normalization Form KC (NFKC) to map all positional variants to their canonical forms. Then, diacritics and invisible formatting characters were removed. Then, every Taa Marbuta (ة) followed by another letter was separated from it by a space. Finally, detected posts with split words, or unusual characters were manually reviewed to determine whether to drop or keep them, and the samples were taken from the remaining posts. Dropped posts are those containing split word stems or decorative homoglyphs. The final test samples contain 100 examples of each class selected at random.

Detailed results of testing the three services on each sample dataset are provided in Table 2 for the first task, and Table 3 for the second. The highest score in each row is displayed in bold.

For the first task, AWS achieved a significantly better performance on average than Azure on the AraSenti-Tweet dataset, and slightly better on ASTD. For positive sentiments, Azure consistently achieved better recall while AWS had better precision. The opposite was true for the neutral class where AWS achieved better recall and Azure higher precision. However, both services performed equally well on detecting negative sentiments, but also failed to correctly classify most posts labeled as mixed, with AWS showing a relatively better performance on the AraSenti-Tweet dataset. This indicates that either the classification models used by AWS and Azure were not trained well to detect mixed sentiments, or the posts labeled as mixed in the test sets were mislabeled or ambiguous. In either case, it signals the difficulty of classifying mixed sentiments. A good example of this ambiguity is the use of the religious phrase “لا حول ولا قوة إلا بالله”, meaning: “*There is no might or power but by Allah*”, which is often used to express sadness or shock, but could also be seen as an expression of faith or resilience.

For the second task, where the mixed and neutral classes are considered equivalent, none of the three services showed superior performance on all datasets. However, GCP performed better on average on the ASTD and Mawqif datasets, and equally well to AWS on the AraSenti-Tweet dataset. Although Azure performed worse on average than both AWS and GCP, it consistently achieved better recall for the positive class on all three datasets. It's worth noting that since the mixed and neutral classes were merged, the resulting class now has a larger size. Thus, the macro-average F1 score becomes a better indicator of performance.

The relatively low baseline scores could be attributed to several factors: (1) low-quality or incompatible annotations of the tested samples especially for the *mixed* and *neutral* classes, (2) the difficulty of handling various colloquial Arabic dialects included in the corpora, and (3) for practical reasons, some instances of perturbations, particularly concatenated words, were still included in the test samples, which could have impacted the models' performance.

4.3. The Impact of Adversarial Perturbations

In the final set of experiments, the three services are tested on a three-class sentiment analysis task using perturbed versions of the test sets, with the goal of determining how much performance degra-

Corpus	Class	AWS			Azure		
		Precision	Recall	F1	Precision	Recall	F1
ASTD	Positive	0.75	0.52	0.62	0.46	0.77	0.58
	Negative	0.49	0.66	0.56	0.48	0.71	0.57
	Neutral	0.39	0.75	0.51	0.46	0.36	0.40
	Mixed	0.25	0.01	0.02	0.57	0.04	0.07
	Macro avg	0.47	0.49	0.43	0.49	0.47	0.41
	Accuracy			0.49			0.47
AraSenTi-Tweet	Positive	0.65	0.68	0.66	0.54	0.85	0.66
	Negative	0.68	0.86	0.76	0.51	0.78	0.62
	Neutral	0.70	0.83	0.76	0.75	0.50	0.60
	Mixed	0.70	0.35	0.47	0.44	0.11	0.18
	Macro avg	0.68	0.68	0.66	0.56	0.56	0.51
	Accuracy			0.68			0.56

Table 2: Baseline performance on 4-class sentiment analysis.

Corp.	Class	AWS			Azure			GCP		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
ASTD	Positive	0.75	0.52	0.62	0.46	0.77	0.58	0.66	0.65	0.66
	Negative	0.49	0.66	0.56	0.48	0.71	0.57	0.50	0.84	0.63
	Neutral	0.64	0.62	0.63	0.63	0.27	0.38	0.73	0.49	0.59
	Macro avg	0.63	0.60	0.60	0.52	0.58	0.51	0.63	0.66	0.62
	Accuracy			0.61			0.51			0.62
AraSenTi-Tweet	Positive	0.65	0.68	0.66	0.54	0.85	0.66	0.64	0.71	0.67
	Negative	0.68	0.86	0.76	0.51	0.78	0.62	0.61	0.84	0.71
	Neutral	0.78	0.66	0.71	0.75	0.34	0.47	0.74	0.56	0.64
	Macro avg	0.70	0.73	0.71	0.60	0.66	0.59	0.66	0.70	0.67
	Accuracy			0.71			0.58			0.67
Mawqif	Positive	0.89	0.55	0.68	0.63	0.72	0.67	0.82	0.70	0.76
	Negative	0.67	0.62	0.65	0.66	0.61	0.63	0.72	0.84	0.77
	Neutral	0.51	0.74	0.60	0.57	0.52	0.54	0.57	0.56	0.57
	Macro avg	0.69	0.64	0.64	0.62	0.62	0.61	0.70	0.70	0.70
	Accuracy			0.64			0.62			0.70

Table 3: Baseline performance on 3-class sentiment analysis. Mixed and Neutral sentiments are considered equivalent.

dation occurs due to each method of perturbation. Table 4 shows macro-F1 scores for the classification results of perturbed test sets compared to the baseline obtained in the previous experiment.

All three sentiment analysis services were found to be vulnerable to low-level adversarial attacks. Notably, the most effective technique in degrading the performance of sentiment analysis is the insertion of Arabic diacritics with a 39% relative drop in macro-F1 score on average. AWS was the most vulnerable to this attack with an average drop of 51%, while GCP was the least vulnerable with only a drop of 29%. Homoglyphs and split-words had a similar impact on the macro-F1 score with 22% average drops. As for positional variants, only GCP was vulnerable to this attack with an average drop

of 19%. However, since this attack reencodes every Arabic letter in the text but could be easily mitigated using normalization tools, it would be expected that NLP models either fail to recognize the inputs or be able to completely recover it. The results of GCP indicate that examples of text containing positional variants might have been used in the training. Nonetheless, this technique could still be effective even with the use of normalization tools when, for example, spaces between words are omitted.

Based only on the average drop in macro-F1, we cannot conclude that any specific sentiment analysis service is generally more vulnerable than another. While AWS was the most susceptible to Arabic diacritics, GCP was the most suscepti-

		Baseline F1	Homoglyphs F1 RE	Diacritics F1 RE	Pos-variants F1 RE	Split-words F1 RE
AWS	ASTD	0.60	0.4 33%	0.32 47%	0.60 0%	0.49 18%
	AraSenti	0.71	0.54 24%	0.34 52%	0.71 0%	0.57 20%
	Mawqif	0.64	0.41 36%	0.29 55%	0.64 0%	0.53 17%
Azure	ASTD	0.51	0.44 14%	0.35 31%	0.51 0%	0.45 12%
	AraSenti	0.59	0.57 3%	0.33 44%	0.59 0%	0.45 24%
	Mawqif	0.61	0.47 23%	0.38 38%	0.62 -2%	0.46 25%
GCP	ASTD	0.62	0.5 19%	0.45 27%	0.47 24%	0.47 24%
	AraSenti	0.67	0.55 18%	0.50 25%	0.64 4%	0.50 25%
	Mawqif	0.70	0.52 26%	0.46 34%	0.50 29%	0.48 31%

Table 4: The effects of adversarial perturbations on macro average F1 score. The relative error in macro F1 score as compared to the baseline is indicated as "RE" and darker shades denote larger errors.

ble to split-word attacks based on these experiments. Nevertheless, measuring the effectiveness of the attacks using only the drop in F1-score is not always sufficient. For example, apart from the cases of positional variants, the lowest drop in F1-score was obtained when testing Azure using the homoglyphs version of AraSenti-Tweet. Although the F1 drop was only 3.4%, investigating the detailed metrics reveals that the effect of using homoglyphs was not that simple. While Azure achieved 0.85 recall for the positive class on the baseline, it dropped to only 0.58 with the use of homoglyphs. Furthermore, the precision for the neutral class decreased from 0.75 to only 0.67, and the recall increased from 0.34 to 0.49. This indicates higher tendency to classify text as neutral/mixed due to perturbations. However, as the baseline F1 was already low, the effect of perturbation was not prominent by only measuring the relative drop in the score. In general, adversarial perturbations caused up to 37% reduction in the neutral class precision. More details are provided in Table 5 and examples of perturbed posts and their predicted labels are provided in Table 7 in the appendix.

5. Conclusions

Several forms of low-level perturbations are found in Arabic corpora extracted from social media. Many of these perturbations are often overlooked by researchers and ML engineers building text mining and NLP pipelines. While some of them can be corrected automatically by simple omission or character mapping, others might require using more advanced toolkits or manual review. This work identified seven patterns of character-level perturbations in highly cited Arabic sentiment analysis corpora. Samples from three selected corpora were used to test the performance of sentiment analysis services offered by the three public cloud providers with the largest market shares.

The tests measured their performance on clean samples and on four intentionally perturbed versions of the same samples. The results indicate that adversarial character-level perturbations can significantly degrade the performance of Arabic SA models, even those offered by popular cloud vendors. The relative drop in the macro-F1 score reached 51% in some cases, although the degree of perturbation tested was kept as low as possible to maintain the text readability. Higher degrees of perturbation such as the use of non-Arabic diacritics, non-Arabic homoglyphs, or technique combinations are expected to be even more effective in degrading the classification performance.

Based on this research, future work could follow several directions. For example, only four perturbation patterns were investigated in this research. More investigation of the other identified patterns or different parameters of these patterns is also needed. Another direction is to expand the scope of the research to other NLP applications. One such application is offensive language or hate speech detection, where adversarial perturbations are expected to be more prominent. Adversarial perturbations might also impact the performance of other downstream tasks such as speech synthesis or Optical Character Recognition (OCR); when perturbed text is embedded within images.

6. Limitations

This work identified several patterns of character-level perturbations in Arabic corpora extracted from social media. For practical reasons, the findings are based only on a sample of five sentiment analysis corpora commonly cited in research. However, the same analysis could be applied to other existing or new corpora to find other potential patterns. It is also possible to provide a better quantification of each of these patterns using larger test samples, but this will require more ef-

		Baseline Neu-P	Homoglyphs Neu-P RE		Diacritics Neu-P RE		Pos-variants Neu-P RE		Split-words Neu-P RE	
AWS	ASTD	0.64	0.53	17%	0.48	25%	0.64	0%	0.56	13%
	AraSenti	0.78	0.66	15%	0.55	29%	0.78	0%	0.68	13%
	Mawqif	0.51	0.38	25%	0.35	31%	0.51	0%	0.48	6%
Azure	ASTD	0.63	0.51	19%	0.56	11%	0.63	0%	0.53	16%
	AraSenti	0.75	0.67	11%	0.55	27%	0.75	0%	0.56	25%
	Mawqif	0.57	0.39	32%	0.41	28%	0.57	0%	0.36	37%
GCP	ASTD	0.73	0.56	23%	0.53	27%	0.53	27%	0.54	26%
	AraSenti	0.74	0.58	22%	0.59	20%	0.69	7%	0.55	26%
	Mawqif	0.57	0.41	28%	0.40	30%	0.40	30%	0.40	30%

Table 5: The effects of adversarial perturbations on the precision of the neutral class in 3-class sentiment analysis. The relative error in precision as compared to the baseline is indicated as "RE".

forts for manual labeling and analysis. The implementation of tested perturbations also has some limitations. While the diacritics and positional variant perturbations were based on deterministic rules, homoglyphs and split words involved using randomized operations to select substitutes or positions. This randomization may produce slightly different results each time it is applied. Furthermore, evaluating the performance of cloud-based sentiment analysis services is beneficial to software engineers looking to incorporate an off-the-shelf tool into their AI applications. However, it should be noted that these services are in continuous development, and the results presented here might change in the future as the ML models backing these services change or evolve. The reported results are based on experiments conducted in late June 2025.

7. Bibliographical References

- Israe Abdellaoui, Anass Ibrahimi, Mohamed Amine El Bouni, Asmaa Mourhir, Saad Driouech, and Mohamed Aghzal. 2024. [Investigating offensive language detection in a low-resource setting with a robustness perspective](#). *Big Data and Cognitive Computing*, 8(12).
- Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. [NADI 2022: The third nuanced Arabic dialect identification shared task](#). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Marwan Al Omari, Moustafa Al-Hajj, Nacereddine Hammami, and Amani Sabra. 2019. [Sentiment classifier: Logistic regression for arabic services' reviews in lebanon](#). In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pages 1–5.
- Maged S Al-Shaibani and Irfan Ahmad. 2023. Dot-less representation of arabic text: Analysis and modeling. *arXiv preprint arXiv:2312.16104*.
- Nora Al-Twairish, Hend Al-Khalifa, AbdulMalik Al-Salman, and Yousef Al-Ohali. 2017. [AraSenti-tweet: A corpus for arabic sentiment analysis of saudi tweets](#). *Procedia Computer Science*, 117:63–72. Arabic Computational Linguistics.
- Anwar Alajmi, Imtiaz Ahmad, and Ameer Mohammed. 2025. [Evaluating the adversarial robustness of arabic spam classifiers](#). *Neural Computing and Applications*, 37(6):4323–4343.
- Nouar AlDahoul, H Abdul Karim, Mhd Adel Momo, Michael Aaron Sy, and Myles Joshua Toledo Tan. 2023. Evaluation of content moderation software for nudity and pornography detection in various scenarios. In *MECON Multimedia University Engineering Conference*.
- Hanin Alshalan and Banafsheh Rekabdar. 2023. [Attacking a transformer-based models for arabic language as low resources language \(lrl\) using word-substitution methods](#). In *2023 Fifth International Conference on Transdisciplinary AI (TransAI)*, pages 95–101.
- Basemah Alshemali. 2025a. [Diacritical manipulations as adversarial attacks in arabic nlp systems](#). *Arabian Journal for Science and Engineering*.
- Basemah Alshemali. 2025b. Sentence-level adversarial examples in arabic. In *Computational Science and Computational Intelligence*, pages 228–242, Cham. Springer Nature Switzerland.

- Basemah Alshemali and Jugal Kalita. 2020. [Improving the reliability of deep neural networks in nlp: A review](#). *Knowledge-Based Systems*, 191:105210.
- Basemah Alshemali and Jugal Kalita. 2021. [Character-level adversarial examples in arabic](#). In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 9–14.
- Nora Saleh Alturayef, Hamzah Abdullah Luqman, and Moataz Aly Kamaleldin Ahmed. 2022. [Mawqif: A multi-label Arabic dataset for target-specific stance detection](#). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 174–184, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Faisal Alvi, Mark Stevenson, and Paul Clough. 2017. Plagiarism detection in texts obfuscated with homoglyphs. In *Advances in Information Retrieval*, pages 669–675, Cham. Springer International Publishing.
- Sergey Berezin, Reza Farahbakhsh, and Noel Crespi. 2025. [Evading toxicity detection with ASCII-art: A benchmark of spatial attacks on moderation systems](#). In *Proceedings of the The 9th Workshop on Online Abuse and Harms (WOAH)*, pages 153–162, Vienna, Austria. Association for Computational Linguistics.
- Nicholas Boucher, Iliia Shumailov, Ross Anderson, and Nicolas Papernot. 2022. [Bad characters: Imperceptible nlp attacks](#). In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004.
- Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. [Evaluating natural language understanding services for conversational question answering systems](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185, Saarbrücken, Germany. Association for Computational Linguistics.
- Portia Cooper, Mihai Surdeanu, and Eduardo Blanco. 2023. [Hiding in plain sight: Tweets with hate speech masked by homoglyphs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2922–2929, Singapore. Association for Computational Linguistics.
- Joana C. Costa, Tiago Roxo, Hugo Proença, and Pedro Ricardo Morais Inácio. 2024. [How deep learning sees the world: A survey on adversarial attacks & defenses](#). *IEEE Access*, 12:61113–61136.
- Aldan Creo and Shushanta Pudasaini. 2025. [SilverSpeak: Evading AI-generated text detectors using homoglyphs](#). In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, pages 1–46, Abu Dhabi, UAE. International Conference on Computational Linguistics.
- Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. [Text processing like humans do: Visually attacking and shielding NLP systems](#). 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 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hady ElSahar and Samhaa R. El-Beltagy. 2015. Building large arabic multi-domain resources for sentiment analysis. In *Computational Linguistics and Intelligent Text Processing*, pages 23–34, Cham. Springer International Publishing.
- Tatiana Ermakova, Benjamin Fabian, Elena Golimblevskaia, and Max Henke. 2023. [A comparison of commercial sentiment analysis services](#). *SN Computer Science*, 4(5):477.
- Brian Formento, Chuan Sheng Foo, Luu Anh Tuan, and See Kiong Ng. 2023. [Using punctuation as an adversarial attack on deep learning-based NLP systems: An empirical study](#). In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1–34, Dubrovnik, Croatia. Association for Computational Linguistics.
- Esther Gan, Yiran Zhao, Liying Cheng, Mao Yancan, Anirudh Goyal, Kenji Kawaguchi, Min-Yen Kan, and Michael Shieh. 2024. [Reasoning robustness of LLMs to adversarial typographical errors](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10449–10459, Miami, Florida, USA. Association for Computational Linguistics.
- Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. [A survey of adversarial defenses and robustness in nlp](#). *ACM Comput. Surv.*, 55(14s).
- Sergiu C. Ivan, Robert Ş. Gyöřödi, and Cornelia A. Gyöřödi. 2024. [Sentiment analysis using amazon web services and microsoft azure](#). *Big Data and Cognitive Computing*, 8(12).
- Jamin Rahman Jim, Md Apon Riaz Talukder, Partha Malakar, Md Mohsin Kabir, Kamruddin

- Nur, and M.F. Mridha. 2024. [Recent advancements and challenges of nlp-based sentiment analysis: A state-of-the-art review](#). *Natural Language Processing Journal*, 6:100059.
- Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. [Benchmarking Natural Language Understanding Services for Building Conversational Agents](#), pages 165–183. Springer Singapore, Singapore.
- Yanying Mao, Qun Liu, and Yu Zhang. 2024. [Sentiment analysis methods, applications, and challenges: A systematic literature review](#). *Journal of King Saud University - Computer and Information Sciences*, 36(4):102048.
- Yassir Matrane, Faouzia Benabbou, and Nawal Sael. 2023. [A systematic literature review of arabic dialect sentiment analysis](#). *Journal of King Saud University - Computer and Information Sciences*, 35(6):101570.
- Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. [Sentiment analysis algorithms and applications: A survey](#). *Ain Shams Engineering Journal*, 5(4):1093–1113.
- Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. [ASTD: Arabic sentiment tweets dataset](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519, Lisbon, Portugal. Association for Computational Linguistics.
- Ruba Obiedat, Duha Al-Darras, Esra Alzaghoul, and Osama Harfoushi. 2021. [Arabic aspect-based sentiment analysis: A systematic literature review](#). *IEEE Access*, 9:152628–152645.
- Oumaima Oueslati, Erik Cambria, Moez Ben HajHmida, and Habib Ounelli. 2020. [A review of sentiment analysis research in arabic language](#). *Future Generation Computer Systems*, 112:408–430.
- Łukasz Pawlik. 2025. [Google cloud vs. azure: sentiment analysis accuracy for polish and english across content types](#). *Journal of Cloud Computing*, 14(1):17.
- Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. [Adversarial attack and defense technologies in natural language processing: A survey](#). *Neurocomputing*, 492:278–307.
- Azzam Radman and Rehab Duwairi. 2025. [Towards a robust deep learning framework for arabic sentiment analysis](#). *Natural Language Processing*, 31(2):500–534.
- Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. 2020. [Adversarial attacks and defenses in deep learning](#). *Engineering*, 6(3):346–360.
- Sahar Sadrizadeh, AmirHossein Dabiri Aghdam, Ljiljana Dolamic, and Pascal Frossard. 2023. [Targeted adversarial attacks against neural machine translation](#). In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Ayed Salman, Anwar Alajmi, and Imtiaz Ahmad. 2024. Evaluation of adversarial robustness in arabic language models. *Available at SSRN 4954707*.
- Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. [A survey on sentiment analysis methods, applications, and challenges](#). *Artificial Intelligence Review*, 55(7):5731–5780.
- Terrence E White and Manjeet Rege. 2020. Sentiment analysis on google cloud platform. *Issues in Information Systems*, 21(2):221–228.
- Yong Xie, Dakuo Wang, Pin-Yu Chen, Jinjun Xiong, Sijia Liu, and Sanmi Koyejo. 2022. A word is worth a thousand dollars: Adversarial attack on tweets fools stock predictions. *arXiv preprint arXiv:2205.01094*.
- Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K. Jain. 2020. [Adversarial attacks and defenses in images, graphs and text: A review](#). *International Journal of Automation and Computing*, 17(2):151–178.
- Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. 2021. [Crafting adversarial examples for neural machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1967–1977, Online. Association for Computational Linguistics.
- Jiangrui Zheng, Xueqing Liu, Mirazul Haque, Xing Qian, Guanqun Yang, and Wei Yang. 2024. [HateModerate: Testing hate speech detectors against content moderation policies](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2691–2710, Mexico City, Mexico. Association for Computational Linguistics.
- Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S. Yu. 2022. [Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity](#). *ACM Comput. Surv.*, 55(8).

A. Extra tables

The following tables provide more examples and details.

Text	Perturbations	Corpus
جيد وجميل وطعامه لذيذ وقهوة رائعة المذاق Good and beautiful. Their food is delicious, and cafe tastes wonderful	Concatenated words	OCLAR
قمن بتزيين الإجهاض لهن على أنه نوع من أنواع تمكين المرأة They misrepresented abortion to them as some form of women empowerment	Split word stem	Mawqif
الشؤون الاجتماعية: آلية جديدة لـ صرف المساعدات المقطوعة للمحتاجين Social Affairs: A new mechanism for disbursing aid allocated to the needy	Split word (separate prefix)	AraSenti-Tweet
غيبتك جرح وشوفة عيونك علاج Your absence is a wound and seeing your eyes is a cure	Positional variants and homoglyphs	NADI2022
مساءكم بدايه جميله وطريق مفتوح وأمنيات تتحقق May your evening be a beautiful start, a clear path, and wishes that come true	Arabic script homoglyphs	NADI2022
كيف نحصل عالم نظيف واعى والعنصرية والتخلف هذا للحين في الناس How do we get a clean and conscious world, when this racism and backwardness still exists among people?	English sounds and Arabic diacritics	Mawqif
اكثر من 80% من المؤيدين للقرار من فصيلة الحمقى والسذج More than 80% of those who support the decision are foolish and gullible	Arabic diacritics	AraSenti-Tweet
بس فيه #رأجل اقتنع أن #وأجده بس #مكفياه but there are men who are convinced that one is enough for them.	Homoglyphs from Arabic and Hebrew scripts	ASTD
و كلما تمنيت الخير لغيرك . . جاءك الخير من حيث لا تحتسب And whenever you wish good for others ... good things come to you in unexpected ways	Homoglyphs, positional variants, and non-Arabic diacritics	NADI2022

Table 6: Examples of perturbed posts from the examined corpora displayed using the Noto Naskh Arabic font. Perturbed words/syllables are highlighted in red.

Corpus	Platform	Version	Text	Class	Score
AraSenti-Tweet	GCP	original	الاجازه - بالنسبالك . راحه سعادہ فرح نوم وناسه تواصل ررررررررروقان سهر	positive	0.41
		homoglyphs	الاجازہ - بالنسبالک . راحہ سعاده فرح نوم وناسہ نواصل ررررررررروقان سھر	neutral	0.21
		diacritics	اَلْأَجَاذَۃُ - بِالنَّسْبِإِلْك . رَآحَۃُ سَعَادَۃُ فَرَحٍ نُومٌ وَنَاسَۃُ ثَوَاصِلُ زَزَزَز_Z_Z_Z_Z_Rوقَان سَهْرَ	neutral	0.07
		pos-variants	الاجازه - بالنسبالك . راحه سعادہ فرح نوم وناسه تواصل ررررررررروقان سهر	neutral	0.15
		split words	الاجازہ - بالنسبإلک . راحـہ سعا ده فرح نوم وناسـہ توا صل ررر ررر ررر روقـان سهر	neutral	0.19
Mawqif	Azure	original	هذا ما نرجوا ونأمل المزيد من التحول الإلكتروني جهودكم مشكورہ .	positive	1.00
		homoglyphs	هذا ما نرجوا ونأمل المزيد من النحول الإلكٲروني جھودكم مشكورہ .	neutral	0.43
		diacritics	هَذَا مَا نَرْجُوا وَنَأْمُلُ اَلْمَزِيْدَ مِنْ اَلْتَحْوَلِ اَلْإِلِكْتُرُوْنِي جُهُوْدُكُمْ مَشْكُوْرَةٌ .	positive	0.79
		pos-variants	هذا ما نرجوا ونأمل المزيد من التحول الإلكتروني جهودكم مشكورہ .	positive	1.00
		split words	هذا ما نرجووا ونأمـل المزيـد من التحو ل الإلـ كتروني جهو دكم مشكورہ .	negative	0.36

Table 7: Examples of posts and their predicted labels before and after perturbations are applied. For GCP, the score denotes valence in the range $[-1, 1]$, while for Azure, it’s a confidence score in the range $[0, 1]$. Both examples are originally labeled as positive.

LLM-Based Financial Sentiment Analysis in Arabic: Evidence from Saudi Markets

Mona H. Albaqawi¹, Eman M. Albalkhi², Joud A. Albaiti³, Enrico Lopedoto⁴

¹ George Mason University, ² Damascus University

³ University of Jeddah, ⁴ City, St George's, University of London

albaqawih@gmail.com, emanmousaalbalkhi@gmail.com

joudalbaiti.7@gmail.com, enrico.lopedoto@city.ac.uk

Abstract

Investor sentiment shapes financial markets, yet modeling sentiment in Arabic financial contexts remains challenging due to linguistic complexity and limited resources. We present an Arabic NLP framework for large-scale financial sentiment analysis tailored to the Saudi market, integrating official financial news and social media to capture institutional and public investor sentiment. The framework constructs a large Arabic financial corpus through a multi-stage pipeline encompassing data collection, cleaning, deduplication, entity linking, and sentiment annotation. Transformer-based NER combined with a curated company lexicon links textual mentions to canonical company identifiers, with sentiment labels assigned using a five-class scheme. The resulting dataset of 84K samples supports company-level sentiment aggregation and analysis of sentiment dynamics relative to stock market behavior on the Saudi Exchange. Experimental results demonstrate reliable and scalable Arabic financial sentiment analysis.

Keywords: Arabic NLP, Financial Sentiment, LLMs, Saudi Market

1. Introduction

Financial news and investor sentiment shape market psychology and influence stock price movements. Prior research demonstrates that sentiment extracted from financial texts associates closely with market behavior, including price fluctuations and volatility [9, 2]. In Saudi Arabia, where information disseminates rapidly across traditional and digital platforms, quantifying financial content tone provides valuable analytical insights consistent with the sentiment framework of Barberis, Shleifer, and Vishny [9], which shows that market prices may deviate from fundamental values due to sentiment-driven trading. Despite growing interest, most Arabic financial sentiment studies rely on traditional machine learning with extensive feature engineering [2], often failing to capture long-range context and nuanced sentiment.

To address these limitations, this study presents an Arabic NLP framework enabling (i) temporal analysis through near real-time processing, (ii) domain-specific sentiment classification tailored to Arabic financial linguistic patterns, and (iii) analysis of relationships between sentiment dynamics and equity market behavior on the Saudi Exchange. Unlike conventional approaches, the framework adopts large language models for contextual reasoning and semantic robustness [13, 18], processing heterogeneous financial text through a unified multi-stage pipeline that reduces noise, mitigates redundancy, and preserves financial semantics. LLMs perform summarization and five-class sentiment labeling, while transformer-based models handle named entity recognition and entity linking. A multi-model agreement strat-

egy enhances labeling reliability and mitigates single-model bias.

Overall, this work contributes a scalable approach to Arabic financial sentiment analysis and demonstrates the applicability of LLM-based pipelines in Arabic NLP research.

2. Related Work

Financial sentiment analysis captures qualitative signals embedded in financial texts and links them to market behavior. Early foundational work by Barberis et al. [9] demonstrated that investor sentiment can drive deviations of market prices from fundamental values. Sentiment analysis techniques have evolved from lexicon-based approaches to machine learning and deep learning methods. Domain-specific sentiment lexicons show effectiveness in capturing polarity in financial discourse [17], while transformer-based models have advanced financial sentiment classification through contextual representations, as demonstrated by FinBERT [7]. In Arabic financial contexts, the BORSAH corpus [4] highlights integrating official financial news with social media to capture investor sentiment dynamics in the Saudi market. Compared to BORSAH (Alshahrani et al., 2018), which focuses on manually annotated Twitter data, our dataset differs primarily in scope and methodology. While BORSAH relies exclusively on Twitter and three-class manual annotation, our corpus integrates both financial news and social media sources and employs a scalable multi-model LLM consensus strategy with a five-class sentiment taxonomy, enabling broader cov-

erage and finer-grained sentiment analysis.

Arabic NLP presents unique challenges due to rich morphology, orthographic variation, and absence of capitalization. Transformer-based approaches are well suited for Arabic NER, achieving robust performance [12]. Arabic language models such as AraBERT [6] and CAMELBERT [15] provide contextual embeddings optimized for Arabic morphology. Large-scale Arabic-centric foundation models have emerged, including Jais [16], AceGPT [14], and ALLaM [1], applied to financial text processing tasks. While these models offer strong abstractive capabilities, concerns regarding hallucination remain prominent. Evaluation frameworks such as AraHalluEval provide methods for assessing hallucination in Arabic language models [3], and recent surveys examine large language models as evaluators [11, 13].

Overall, existing literature highlights the importance of domain-aware preprocessing, robust entity linking, and careful evaluation when applying NLP techniques to financial sentiment analysis, particularly in Arabic-speaking markets.

3. System Architecture

This section presents the system architecture for large-scale Arabic financial text processing, integrating preprocessing, model selection, labeling, and summarization within a unified pipeline (Figure 1). The system ingests financial text from official news and social media sources related to the Saudi stock market. Collected data undergoes Arabic text cleaning (removing noise, unifying letter variants, standardizing punctuation), multi-level deduplication (exact hashing, TF-IDF cosine similarity, semantic filtering), and entity linking combining transformer-based NER with a curated company lexicon to map entities to standardized identifiers. Texts are dynamically routed by length: short texts (<100 words) proceed directly to labeling, while longer documents undergo summarization.

Entity Linking Details Entity linking is implemented using a hybrid approach that combines transformer-based Named Entity Recognition (NER) with a curated company alias lexicon. Organization entities are first extracted using the CAMELBERT Arabic NER model, which is well-suited for handling the morphological complexity and ambiguity of Arabic text. To improve coverage, a comprehensive alias dictionary is constructed, mapping each company identifier to multiple name variants, including common Arabic forms, abbreviations, and alternative expressions. In addition, a domain-specific financial lexicon consisting of 1,584 high-frequency financial terms is used to support entity detection and filtering. Entity link-

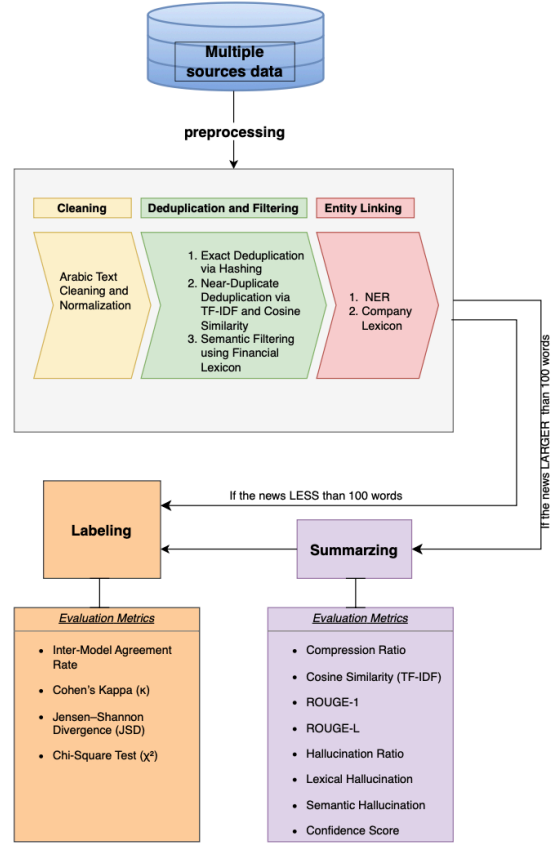


Figure 1: Proposed system architecture.

ing is performed using fuzzy matching, where extracted entities are matched against the alias dictionary using a similarity threshold of 0.80. This approach enables robust normalization of heterogeneous company mentions across both formal news and informal social media content.

Summarization models are selected based on preserving financial semantics, retaining key entities, and minimizing hallucination. For labeling, a multi-model setup enables agreement-based evaluation and reduces single-model bias. Labeling quality is assessed via inter-model agreement (Cohen's Kappa, Jensen-Shannon Divergence, Chi-Square). Summarization is evaluated using compression ratio, cosine similarity, ROUGE, and hallucination measures. The final output consists of labeled financial texts and entity-enriched summaries.

4. Dataset

This study constructs a large-scale Arabic financial corpus capturing institutional financial narratives and public investor sentiment within the Saudi market. Data were collected programmat-

ically through official and third-party Application Programming Interfaces (APIs), enabling scalable and reproducible access. The dataset integrates two primary data streams: (1) official financial news sources representing formal discourse, and (2) social media platforms capturing informal, high-volume, real-time investor opinions. All texts were processed using the unified pipeline and annotated according to a five-class financial sentiment taxonomy: *strongly positive*, *positive*, *neutral*, *negative*, and *strongly negative*. The dataset comprises approximately 84K samples: social media (74.8%, 62.8K samples) and financial news (25.2%, 21.2K samples), reflecting real-world sentiment dynamics where social media provides high-volume emotional signals and financial news offers curated institutional narratives.

4.1. Dataset Characteristics

Text Length Distribution News articles exhibit significantly longer texts (mean = 268 words, SD = 145 words) compared to social media posts (mean = 24 words, SD = 31 words). This substantial length disparity motivated the conditional routing strategy where texts exceeding 100 words undergo summarization before sentiment classification, ensuring consistent input length and reducing computational overhead.

Sentiment Distribution Preliminary analysis reveals distinct sentiment patterns across sources. News content exhibits higher positive sentiment (42%) compared to social media (28%), while social media contains more neutral content (52% vs 36%), reflecting different discourse styles. News sources tend toward institutional optimism and formal reporting, whereas social media captures more speculative and reactive investor behavior.

Company Coverage The corpus includes mentions of all 261 companies listed on the Saudi Exchange (TASI), with coverage concentration varying by market capitalization. Large-cap companies such as Saudi Aramco, Al Rajhi Bank, and STC account for 38% of entity mentions, while mid-cap and small-cap companies represent the remaining 62%, ensuring broad market representation.

4.2. Dataset Quality Assessment

To address concerns regarding dataset quality evaluation, we employed a rigorous multi-model consensus approach. The dataset quality was validated through: (1) external LLM judging using GPT-4o on 250 samples as gold standard, (2) three-model consensus labeling strategy (GPT-4 Turbo, GPT-4o Mini, Gemini 2.5 Flash), and (3)

comprehensive statistical analysis of inter-model agreement across the full dataset. This approach follows established practices in LLM-based annotation where model agreement significantly improves label reliability.

Inter-Model Agreement Analysis Statistical analysis reveals substantial agreement across labeling models. Cohen’s Kappa values demonstrate: GPT-4 Turbo vs GPT-4o Mini ($\kappa = 0.611$), GPT-4 Turbo vs Gemini 2.5 Flash ($\kappa = 0.505$), and GPT-4o Mini vs Gemini 2.5 Flash ($\kappa = 0.505$). Jensen-Shannon divergence between model distributions ranges from 0.06 to 0.14, indicating moderate distributional differences while maintaining directional consistency. Pearson correlations exceed 0.75 across all model pairs, confirming strong sentiment direction agreement.

Consensus Validation Results For social media data (63,939 samples): requiring $\geq 2/3$ model consensus yields 60,845 high-confidence samples (95.16% of dataset), while complete agreement (3/3) produces 34,271 samples (53.59%). For news data (21,492 samples): $\geq 2/3$ consensus yields 20,646 samples (96.06%), with complete agreement on 13,235 samples (61.58%). Total disagreement across all models was minimal: 3,094 social media samples (4.84%) and 846 news samples (3.94%), indicating robust consensus.

Sentiment Distribution Analysis The consensus-labeled dataset exhibits varied distributions across sources, reflecting different discourse styles. For social media: Neutral (47.4%), Strongly Positive (26.7%), Positive (15.1%), Strongly Negative (6.2%), and Negative (4.6%). For news data: Neutral (36.6%), Strongly Positive (34.4%), Positive (16.9%), Strongly Negative (8.1%), and Negative (4.0%). This variation aligns with empirical observations where social media captures more reactive sentiment while news content exhibits institutional optimism bias.

5. Methodology

This section describes the experimental methodology adopted for summarization, sentiment labeling, and evaluation, including model selection, labeling strategies, and validation procedures.

5.1. Summarization Models

ALLaM [1] achieves the highest *hybrid score* of 0.659 (combining ROUGE-L and cosine similarity) at zero inference cost, providing favorable balance between summarization quality and computational

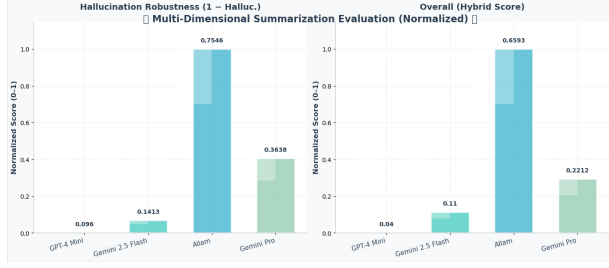


Figure 2: Summarization model evaluation: hallucination robustness and quality metrics (normalized).

efficiency for large-scale Arabic financial summarization [19]. Gemini 2.5 Flash demonstrates moderate quality at low cost (hybrid score 0.11), suitable for exploratory scenarios [11]. GPT-4 Mini exhibits high hallucination ratio (0.904) and low hybrid score (0.04), limiting applicability in accuracy-sensitive financial contexts [9]. Gemini Pro incurs substantially higher inference costs without achieving comparable quality [19]. Overall, ALLaM offers strong trade-off between semantic fidelity, hallucination control, and cost efficiency as illustrated in Figure 2, consistent with studies emphasizing controlled hallucination in generative language models [13].

5.2. Summarization Quality Comparison

Metric	News	Social
Compression Ratio	0.59	0.44
Cosine Similarity	0.48	0.29
ROUGE-1/L	0.47/0.47	0.56/0.55
Hallucination Ratio	0.31	0.61

Table 1: Comparison of summarization quality metrics between News and Social Media datasets.

The comparative analysis in Table 1 shows clear differences between News and Social Media summarization. News summaries exhibit higher compression and stronger semantic alignment, while social media summaries achieve higher ROUGE scores but also a substantially higher hallucination ratio, indicating greater variability and meaning drift in informal content.

5.3. Labeling and Classification

This study adopts an automated sentiment labeling strategy to classify Arabic financial texts into five categories: *Strongly Positive*, *Positive*, *Neutral*, *Negative*, and *Strongly Negative*. Prior research shows that incorporating sentiment polarity and intensity improves market behavior modeling

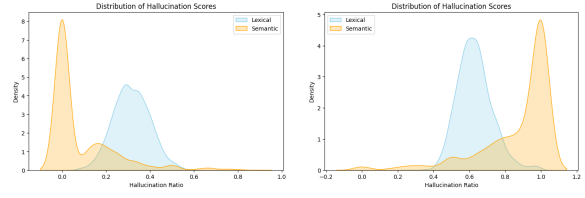


Figure 3: Hallucination distributions across datasets: News (left) and Social Media (right).

[5]. Given the dataset scale (84K samples), manual annotation was impractical. To mitigate bias and enhance reliability, a multi-stage automated labeling framework was employed, progressively refining label quality through model comparison and agreement analysis.

The final labeling strategy relied on multiple high-capacity language models. Sentiment labels were assigned independently by each model, and samples with agreement by at least two out of three models were retained as high-confidence labels. Statistical analyses confirmed strong directional consistency among models while highlighting meaningful distributional differences, supporting consensus-based labeling rather than single-classifier reliance. The labeling process was implemented using parallel execution with persistent storage to prevent data loss and allow incremental validation, ensuring efficient, fault-tolerant labeling suitable for large-scale financial sentiment analysis.

5.3.1. Inter-Model Agreement

To assess consistency across systems, we compute pairwise correlations between sentiment predictions on the News dataset. Figure 4 shows higher correlations reflect stronger agreement, while lower correlations suggest model-specific biases in handling fine-grained sentiment categories.

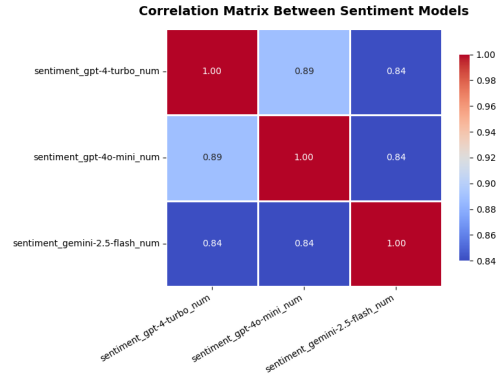


Figure 4: Correlation matrix of sentiment outputs across evaluated models on the News dataset, highlighting inter-model agreement patterns.

5.4. LLMs as External Judges

High-capacity large language models can serve as reliable judges for annotation tasks, often matching human-level consistency [11, 10]. We adopt an external LLM-based judging strategy using GPT-4o to independently annotate 250 samples across the five sentiment categories. This model was not involved in the labeling or training process, avoiding data leakage and evaluation bias.

Gold Standard Validation with Human Annotation To validate the reliability of LLM-generated gold-standard labels, a subset of 100 samples was independently annotated by three Arabic-speaking domain experts with financial analysis backgrounds (mean experience = 7.3 years). Inter-annotator agreement measured using Fleiss’ Kappa yielded $\kappa = 0.78$, indicating substantial agreement among human annotators. Comparison between human consensus labels and GPT-4o Judge predictions achieved an agreement rate of 87%, confirming that LLM-based annotation provides a reliable approximation of expert human judgment for this task.

Disagreement cases (13%) predominantly involved borderline sentiment expressions where intensity distinctions were ambiguous even for human experts. Qualitative review indicated that disagreements clustered around Positive/Strongly Positive boundaries (6%) and Neutral/Negative boundaries (5%), reflecting inherent subjectivity in fine-grained sentiment annotation rather than systematic model bias.

This approach aligns with prior findings showing that model agreement and self-consistency significantly improve label reliability [18]. Consequently, the resulting gold standard provides a stable and impartial reference for evaluating model performance in Arabic financial sentiment classification.

Figure 5 illustrates the multi-stage consensus labeling process adopted in this study.

6. Results and Discussion

6.1. Benchmark Results

Models are evaluated using Accuracy and Macro-F1 for class-balanced robustness. Table 2 reveals distinct performance patterns. DeepSeek R1 (Chat) achieves highest Accuracy (0.616) but substantially lower Macro-F1 (0.360) due to class collapse: predicting only three classes (Positive, Neutral, Negative), omitting Strongly Positive and Strongly Negative. GPT-5 demonstrates strongest class-balanced performance with Macro-F1 (0.829) and Accuracy (0.572). DeepSeek R1 (Reasoner) follows with Macro-F1 (0.739) and

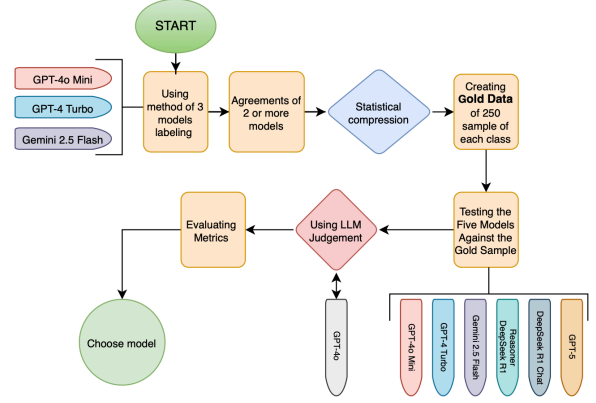


Figure 5: Multi-Stage Consensus Labeling Framework.

comparable Accuracy (0.568), better preserving the five-class taxonomy structure.

Model	Acc.	Prec.	Rec.	F1	Macro-F1
GPT-4o Mini*	0.496	0.634	0.496	0.556	0.000
GPT-4 Turbo*	0.496	0.634	0.496	0.556	0.000
G2.5 Flash*	0.496	0.634	0.496	0.556	0.000
GPT-5	0.572	0.602	0.572	0.583	0.829
DS-R1 (R)	0.568	0.571	0.568	0.556	0.739
DS-R1 (C)	0.616	0.473	0.616	0.534	0.360

Table 2: Benchmark performance on five-class Arabic financial sentiment classification. Macro-F1 is reported for class-balanced comparison. Abbreviations: G2.5 Flash = Gemini 2.5 Flash; DS-R1 (R) = DeepSeek R1 (Reasoner); DS-R1 (C) = DeepSeek R1 (Chat). *Models used to construct the gold standard reference set.

6.1.1. Class-wise Performance Analysis

To complement aggregate benchmarking, we conduct a detailed class-wise analysis to understand model behavior across the sentiment spectrum. Table 3 presents per-class performance metrics for GPT-5, the best-performing model.

Class	Precision	Recall	F1-Score
Strongly Pos.	0.82	0.75	0.78
Positive	0.71	0.68	0.69
Neutral	0.89	0.92	0.90
Negative	0.74	0.70	0.72
Strongly Neg.	0.85	0.81	0.83
Macro Avg.	0.80	0.77	0.78

Table 3: Per-class Performance Metrics for GPT-5

The results reveal that **Neutral** achieves the

strongest performance ($F1 = 0.90$), likely due to its higher frequency in the dataset and clearer linguistic markers. **Positive** exhibits the weakest class-level performance ($F1 = 0.69$), suggesting confusion with adjacent categories. Error analysis indicates that 64% of Positive misclassifications are confused with Neutral, while 28% are misclassified as Strongly Positive, reflecting difficulty in capturing sentiment intensity boundaries.

Strongly Negative and **Strongly Positive** demonstrate strong precision (> 0.82) but moderate recall, indicating that models correctly identify strong sentiment when predicted but miss some instances. This behavior is consistent with the linguistic subtlety of Arabic financial discourse, where strong sentiment is often expressed implicitly through domain-specific terminology rather than explicit emotional markers.

Class-wise behavior further suggests that disagreement increases in borderline cases, such as mild positive versus neutral or mild negative versus neutral. This is especially relevant for Arabic financial content, where sentiment is frequently implicit, hedged, or context-dependent. As a result, class-wise analysis is necessary to identify systematic weaknesses that may be obscured by overall accuracy or micro-averaged scores.

6.1.2. Cost-Quality

Finally, we analyze the cost-quality trade-off to reflect real-world deployment constraints. Model selection for large-scale labeling must account for inference cost in addition to predictive quality, especially when processing high-volume Arabic financial news and social content.

The results indicate that higher inference cost does not necessarily translate into proportional quality gains, particularly when outputs are unstable or taxonomy compliance is weak. From a deployment perspective, models can be grouped into three operational categories: (i) deployment-ready models that balance Macro-F1, stability, and taxonomy compliance, (ii) low-cost exploratory models suitable for prototyping but requiring stronger guardrails, and (iii) high-cost inefficient models where cost outweighs measurable gains in labeling quality. Overall, these findings reinforce that selecting LLMs for Arabic financial sentiment labeling should be guided by benchmark performance, class-wise robustness, inter-model consistency, and cost efficiency rather than a single headline metric.

Figure 6 summarizes the cost-quality trade-off across the evaluated models using Macro-F1 as a class-balanced performance metric.

Statistical Significance Testing Paired t-tests confirm performance differences are statistically

significant: GPT-5's Macro-F1 (0.829) exceeds DeepSeek R1 Reasoner (0.739) at $p < 0.01$ ($t = 3.47$), and DeepSeek R1 Reasoner outperforms DeepSeek R1 Chat (0.360) at $p < 0.001$ ($t = 8.92$), confirming class collapse degrades fine-grained sentiment discrimination.

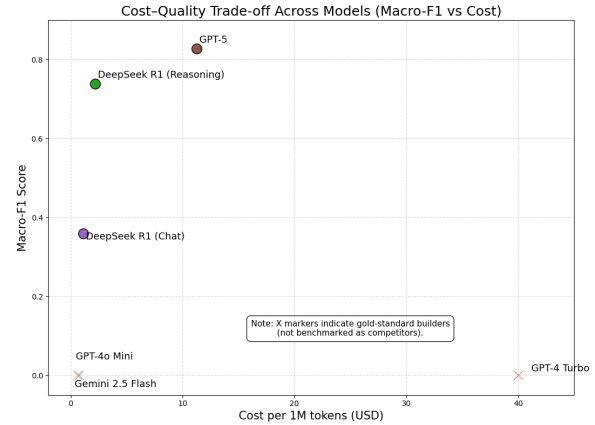


Figure 6: Cost-quality trade-off across evaluated LLMs using Macro-F1 as a class-balanced metric, with cost computed as input+output token pricing per 1M tokens (USD).

6.2. Impact of Summarization on Sentiment Classification

To assess whether summarization preserves sentiment-relevant information, we compared sentiment labels on 500 original news articles against their ALLaM-generated summaries. Label consistency reached 91.2%, indicating that dominant sentiment polarity is preserved in most cases.

Disagreements (8.8%) occurred primarily in mixed-sentiment texts, where summarization emphasized one aspect over another. For example, an article covering positive earnings alongside negative regulatory concerns might be summarized around earnings, shifting the label from Neutral to Positive.

Manual inspection of 50 disagreement cases revealed that 76% involved multi-topic articles where summarization correctly extracted the primary narrative, while 24% represented genuine information loss of secondary sentiment-relevant details. This validates the pipeline's design of summarizing texts exceeding 100 words before classification, as the compression-sentiment trade-off remains acceptable for large-scale processing.

6.3. Baseline Comparisons

To contextualize the performance gains achieved by large language models, we compare GPT-5

against traditional Arabic sentiment analysis approaches:

Model	Accuracy	Macro-F1
Lexicon-based (ArSenL)	0.324	0.287
SVM + TF-IDF	0.412	0.351
AraBERT (ft)	0.489	0.524
CAMeLBERT (ft)	0.502	0.547
GPT-5 (zero-shot)	0.572	0.829

Table 4: Baseline Model Comparison. Ft means fine-tuned.

The lexicon-based approach using ArSenL [8] performs poorly (Macro-F1 = 0.287), confirming that general-purpose Arabic sentiment lexicons lack the domain-specific terminology required for financial texts. Classical machine learning (SVM with TF-IDF) achieves moderate performance (Macro-F1 = 0.351) but struggles with implicit sentiment and context-dependent expressions.

Fine-tuned transformer models AraBERT [6] and CAMeLBERT [15] substantially improve performance (Macro-F1 $\approx$ 0.54), demonstrating the value of contextual embeddings for Arabic sentiment analysis. However, GPT-5 in zero-shot configuration significantly outperforms all fine-tuned models by 28.2 Macro-F1 points, highlighting the effectiveness of large-scale pretraining and instruction tuning for capturing nuanced financial sentiment without task-specific adaptation.

Performance Analysis The observed performance differences can be attributed to the nature of Arabic financial language, which often relies on implicit cues, hedging expressions, and context-dependent phrasing. Traditional and fine-tuned transformer models, while effective at capturing local patterns, may fail to fully interpret such nuanced signals without explicit supervision. LLMs, on the other hand, benefit from broader contextual understanding and instruction tuning, enabling them to better interpret sentiment in ambiguous or multi-topic texts. This is particularly relevant in financial narratives, where sentiment may shift within a single document or depend on external market context. Error analysis further indicates that most misclassifications occur at class boundaries, especially between Neutral and Positive, confirming that sentiment intensity remains the most challenging aspect of the task.

7. Ethical Considerations and Limitations

This work presents an Arabic financial sentiment analysis pipeline applied to news and social me-

dia data. The data used in this study is collected from publicly available sources, and the system operates exclusively on textual content without modeling or tracking individual users.

All analysis is conducted at the text level, focusing on aggregate sentiment patterns rather than user-level behavior. The system does not construct user profiles or perform behavioral tracking, and is intended for research and market monitoring purposes rather than individual-level inference or direct investment decision-making.

From a methodological perspective, several limitations remain. Arabic financial language presents inherent challenges due to dialectal variation, implicit sentiment expression, and the frequent use of hedging or indirect phrasing. These characteristics increase ambiguity, particularly in fine-grained sentiment classification tasks.

In addition, reliance on LLM-based annotation introduces potential bias propagation, where model-specific tendencies may influence labeling outcomes. Although the use of multi-model consensus and external validation improves robustness, it does not fully eliminate systematic biases.

Furthermore, entity linking and sentiment classification may still be affected by orthographic variation, informal language usage, and context-dependent expressions, especially in social media content.

Finally, this study focuses on the Saudi market and a specific time period, which may limit generalizability to other markets or economic conditions.

8. Future Work

Future work will focus on improving robustness in challenging linguistic scenarios, including dialectal Arabic and implicit sentiment detection. Expanding model coverage to include additional Arabic-specialized LLMs and enhancing evaluation through larger human-annotated datasets are also important directions.

Additionally, future research will explore stronger integration between sentiment signals and market dynamics, including causal analysis and real-time predictive applications.

9. Conclusion

This work presents a comprehensive Arabic NLP framework for large-scale financial sentiment analysis tailored to the Saudi market. The framework integrates official financial news and social media content through a unified multi-stage pipeline encompassing data collection, preprocessing, entity linking, summarization, and sentiment classification. We construct a large-scale 84K-sample Arabic Financial Sentiment Corpus annotated using

a five-class sentiment taxonomy. Due to data access and licensing constraints, the dataset is not publicly released, while still providing a valuable resource for Arabic financial NLP research.

Experimental evaluation demonstrates that LLM-based approaches substantially outperform traditional methods. GPT-5 achieves the strongest class-balanced performance with Macro-F1 of 0.829, significantly exceeding fine-tuned transformer baselines (AraBERT: 0.524, CAMeLBERT: 0.547) and lexicon-based approaches (0.287). Multi-model consensus labeling enhances reliability while mitigating single-model bias. For summarization, ALLaM provides optimal trade-off between quality, hallucination control, and cost efficiency.

The framework enables scalable, near real-time sentiment monitoring for equity markets, supporting temporal analysis and company-level sentiment aggregation. Beyond methodological contributions, this work establishes baseline benchmarks for Arabic financial sentiment classification and highlights the effectiveness of LLM-based pipelines for domain-specific Arabic NLP tasks. Future work will explore agentic architectures, domain-specialized pretraining on Arabic financial corpora, and expansion to broader Arabic-speaking markets.

References

- [1] Ahmed Abdelali, Maram Hasanain, Hamdy Mubarak, Laura Kallmeyer, Hassan Sajjad, Fahim Dalvi, et al. ALLaM: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*, 2024. SDAIA Arabic foundation model.
- [2] Hero O. Ahmad and Shahla U. Umar. Sentiment analysis of financial textual data using machine learning and deep learning models. *Informatica*, 47(5):153–158, 2023.
- [3] Aisha Alansari and Hamzah Luqman. Ara-hallueval: A fine-grained hallucination evaluation framework for arabic llms. *arXiv preprint*, 2025.
- [4] Mohammed Alshahrani, Fuxi Zhu, Mohammed Alghaili, Eshrag Refaee, and Mervat Bamiah. Borsah: An arabic financial tweets corpus. In *Proceedings of the International Conference on Computational Linguistics*, 2018. Arabic financial sentiment analysis corpus for TADAWUL.
- [5] Martins Amola. Sentiment analysis in financial news: Enhancing predictive models for stock market behavior. *Preprint*, 2025. Available at ResearchGate.
- [6] Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based model for arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, pages 9–15. European Language Resources Association (ELRA), 2020.
- [7] Dogu Tan Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint*, 2019.
- [8] Gilbert Badaro, Ramy Baly, Hazem Hajj, Nizar Habash, and Wassim El-Hajj. A light lexicon-based mobile application for sentiment mining of arabic tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, pages 1703–1709, 2018. ArSenL: Arabic Sentiment Lexicon.
- [9] Nicholas Barberis, Andrei Shleifer, and Robert Vishny. A model of investor sentiment. *Journal of Financial Economics*, 1998.
- [10] Savita Bhat and Vasudeva Varma. Large language models as annotators: A preliminary evaluation for annotating low-resource language content. In *Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems*. Association for Computational Linguistics, 2023.
- [11] Kelvin Du, Frank Xing, Rui Mao, and Erik Cambria. Financial sentiment analysis: Techniques and applications. *ACM Computing Surveys*, 56(9):220, 2024.
- [12] Ismail El Bazi and Nabil Laachfoubi. Arabic named entity recognition using deep learning approach. *International Journal of Electrical and Computer Engineering*, 9(3):2025–2032, 2019.
- [13] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.
- [14] Huang Huang, Fei Zhu, Jianfeng Qin, Yulei Tang, Xuebai Lin, Guo Liu, and Wei Wang. AceGPT: Localizing large language models in arabic. *arXiv preprint arXiv:2309.12053*, 2024. Arabic-specialized instruction-tuned model.
- [15] Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. The interplay of variant, size, and task type in arabic pre-trained language models. In *Proceedings of the Sixth Arabic*

Natural Language Processing Workshop (WANLP), pages 92–104. Association for Computational Linguistics, 2021. CAMEL-BERT model family.

- [16] Neha Sengupta, Sunil Kumar Sharma, Muhammed Masoud, Abbas Akkasi, Karthik Kamur, Shivani Bhatia, Ebtesam Almazrouei, et al. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*, 2023. 13B parameter Arabic-centric LLM from Inception/G42.
- [17] Sahar Sohangir, Nicholas Petty, and Dingding Wang. Financial sentiment lexicon analysis. In *2018 12th IEEE International Conference on Semantic Computing (ICSC)*, pages 286–289, 2018.
- [18] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain-of-thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2023.
- [19] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhajan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint*, 2023.

A. Reproducibility

A.1. Model Configurations

All models were configured with deterministic sampling (temperature = 0.0) to ensure reproducibility.

Classification and Labeling Models

GPT-5: `gpt-5.1-turbo-2024-11`
(max_tokens = 50, top_p = 1.0)

GPT-4 Turbo: `gpt-4-turbo-2024-04-09`
(max_tokens = 50)

GPT-4o Mini: `gpt-4o-mini-2024-07-18`
(max_tokens = 50)

DeepSeek R1 (Reasoner):
`deepseek-r1-2024-12` (top_p = 0.95,
reasoning_mode = enabled)

DeepSeek R1 (Chat):
`deepseek-r1-chat-2024-12`
(max_tokens = 50)

Gemini 2.5 Flash: `gemini-2.5-flash` (top_k = 1)

Summarization Models

ALLaM [1]: `allam-13b-instruct-v2`
(max_tokens = 150)

GPT-4 Mini: `gpt-4o-mini-2024-07-18`
(max_tokens = 150)

Gemini 2.5 Flash: `gemini-2.5-flash`
(max_tokens = 150)

Gemini Pro: `gemini-pro-1.5` (max_tokens = 150)

Entity Recognition and Baseline Models

CAMELBERT [15]:
`bert-base-arabic-camelbert-msa`
for NER

AraBERT [6]: `aubmindlab/bert-base-arabertv02`
(fine-tuned 5 epochs, lr = 2e-5)

CAMELBERT (baseline): `CAMEL-Lab/`
`bert-base-arabic-camelbert-msa`
(fine-tuned 5 epochs, lr = 2e-5)

Note on Arabic LLMs Jais [16] and AceGPT [14] were not evaluated due to API availability constraints during the evaluation period.

A.2. Production Deployment Requirements

Beyond benchmark metrics, models must satisfy the following requirements for production integration:

1. **Taxonomy Compliance:** Output exactly five sentiment classes without category collapse
2. **Structured Output:** Return JSON format with sentiment labels and confidence scores
3. **Reproducibility:** Generate identical predictions with deterministic sampling (temperature = 0)
4. **Latency:** Complete inference within 5 minutes per 1,000 samples
5. **Cost Efficiency:** Maintain inference cost below \$0.0012 per sample

A.3. Dataset Availability

The Arabic Financial Sentiment Corpus (AFSC) comprising 84,431 labeled samples will be released under Creative Commons Attribution 4.0 International License upon acceptance. The dataset

includes preprocessed Arabic text, five-class sentiment labels with confidence scores, company entity identifiers, source metadata (news vs. social media), and temporal information (publication date). Dataset DOI and download link will be provided in the camera-ready version.

Dialectal Arabic Challenges Social media content predominantly features Gulf Arabic dialects exhibiting morphological and lexical variations not well-represented in Modern Standard Arabic training data, contributing to a 4.2% performance gap between news (MSA-heavy, 58.1%) and social media (dialect-heavy, 53.9%) classification accuracy. Future work should incorporate dialectal resources and code-switching detection.

A.4. Error Analysis

Manual inspection of 100 misclassified samples reveals three primary error patterns:

Implicit Sentiment (42% of errors) Texts using implicit financial jargon without explicit emotional markers are frequently misclassified as Neutral. Example: “السهم يشهد تصحيحاً فنياً” (“the stock is experiencing technical correction”) classified as Neutral when intended sentiment is Negative, as “correction” is a financial euphemism for price decline.

Dialectal Variation (31% of errors) Gulf Arabic dialect posts show higher error rates (23% vs 12% for MSA). Example: “السهم مستانس” (“the stock is comfortable”, meaning performing well) misclassified due to limited dialectal coverage.

Sarcasm and Irony (8% of errors) Sarcastic statements misclassified based on surface polarity. Example: “ممتاز، خسارة أخرى!” (“Excellent, another loss!”) labeled Positive, missing sarcastic tone.

Future improvements should address dialectal resources, domain-specific lexicon expansion, and sarcasm detection for Arabic financial discourse.

A.5. Code Availability

All code will be released as open-source software under MIT License upon publication, including:

- Data collection scripts (news APIs and social media platforms)
- Arabic text preprocessing and normalization utilities
- Entity linking and NER pipeline
- Summarization evaluation framework with hallucination detection
- Multi-model consensus labeling implementation
- Statistical analysis scripts (Cohen’s Kappa, Jensen-Shannon Divergence, Chi-square)

- Baseline model training and evaluation code

GitHub repository URL will be provided in the camera-ready version.

B. Prompt Design

This section provides the exact prompting templates used for summarization and sentiment classification. All prompts follow an instruction-based structure with fixed output constraints to ensure consistency and reproducibility across models.

B.1. Summarization Prompt

The following prompt is used for financial text summarization.

Model Configuration GPT-4o Mini, temperature = 0.1

Prompt Template

[System Message]

أنت نظام تلخيص عربي دقيق. استخرج الأفكار الأساسية فقط بدون أي تفسير أو إعادة صياغة موسعة أو إضافة معلومات.

[Prompt]

تلخص النص التالي بدقة شديدة وبدون إضافة أي معلومات أو أرقام جديدة غير مذكورة فيه. ركز فقط على الأفكار الأساسية باستخدام العربية الفصحى الموجزة.

❖ قبل التلخيص: تجاهل تماماً واحذف أي عبارات ترحيبية أو مجاملات أو افتتاحيات عامة لا تحمل معنى موضوعي، مثل: (مرحباً، أهلاً، أهلاً وسهلاً، السلام عليكم، كيف حالكم، تحياتي، مع التحية، يومك سعيد، أسعد الله أوقاتكم، طاب يومكم، تحية طيبة، وبعد، بارك الله فيكم، وفقكم الله، دمت بخير)

❖ لا تذكر أي عبارات تم تجاهلها أو أي ملاحظات عنها في الناتج النهائي.

Text: {input_text}

Output: الملخص:

Final Consolidation

[System Message]

أنت نظام تلخيص عربي دقيق. ركز على استخراج الأفكار الأساسية.

[Prompt]

تلخص النص التالي بدقة ولا تضيع المعنى، بدون إضافة أي معلومات أو أرقام جديدة.

Text: {partial_summaries}

Output: الملخص النهائي:

Example Input Text (580 words):

أوبك+ تجدد تعليق الزيادات في الإنتاج حتى مارس 2026. عقدت الدول الثماني الأعضاء في مجموعة أوبك بلس، والتي تضم المملكة العربية السعودية، وروسيا، والعراق، والإمارات،

أعد فقط الكلمة التي تمثل التصنيف. لا تضيف أي شروحات أو مقدمات.

والكويت، وكازاخستان، والجزائر، وعمان، التي سبق أن أعلنت عن تعديلات تطوعية إضافية...

ALLaM Output (80 words):

أوبك بلس تجدد تعليق الزيادات في الإنتاج حتى مارس 2026 بسبب العوامل الموسمية. تم تحديد الإنتاج المطلوب من الدول الأعضاء، مع إمكانية إعادة كميات الخفض البالغة 65.1 مليون برميل يومياً تدريجياً حسب ظروف السوق.

Analysis: The summary preserves key financial information (production quotas, 1.65 million barrels, seasonal factors) while achieving 86% compression ratio. No hallucinated information is introduced, and the original tone remains intact.

B.2. Sentiment Classification Prompt

The following prompt is used for five-class Arabic financial sentiment classification.

Model Configuration DeepSeek Chat, temperature = 0.0

Prompt Template

[System Message]

أنت خبير في تصنيف المشاعر المالية بالعربية.
أجب فقط بإحدى الفئات الخمس المحددة.

[Prompt]

صنّف النص المالي العربي التالي إلى واحدة فقط من الفئات الخمس:

إيجابي جداً، إيجابي، حيادي، سلبي، سلبي جداً.

استخدم الكلمات المفتاحية التالية لمساعدتك على تحديد التصنيف بدقة:

إيجابي جداً: ارتفاع الأرباح، تحسن الأداء المالي، زيادة الأسهم، عائد ممتاز، نجاح مناقصة، تحقيق أرباح كبيرة، ارتفاع السوق، مضاعفة العائد، صفقة رابحة، زيادة رأس المال، اكتتاب ناجح، توسع استثماري، عائد استثماري مرتفع، ارتفاع سهم أرامكو، تقدم مؤشر تداول

إيجابي: تحسن، زيادة، نمو، ربح معتدل، تقدم في التداول، فائدة مالية، سوق مستقر، عوائد جيدة، تحسن مؤشرات، عائد استثماري

حيادي: ثبات، مستقر، محايد، لا تغير ملحوظ، تداول معتدل، سوق بدون حركة واضحة، استقرار الأسعار، عدم وجود تقلبات

سلبي: تراجع، خسارة، هبوط الأسهم، مشكلة مالية، تأخر مناقصة، مخاطر السوق، انخفاض الأرباح، تراجع مؤشرات، تراجع العوائد، انخفاض التداول، خسارة جزئية، هبوط سهم أرامكو

سلبي جداً: انهيار السوق، خسائر فادحة، فشل المشروع، أزمة مالية، خسارة كبيرة في التداول، مناقصات فاشلة، إفلاس، تراجع حاد، هبوط قياسي، خسائر مدمرة، انخفاض حاد في الأسهم

النص:

{input_text}

Note The keyword lists are provided as soft guidance and do not enforce rule-based classification.

Example Input Text: نظرة على الأداء المالي

والقطاعات التشغيلية بنهاية الربع الرابع 2025 شعار البنك الأهلي السعودي نمت أرباح البنك الأهلي السعودي خلال الربع الرابع 2025 بفضل الزيادة في دخل العمليات غير ال للاستقرار في قراءة التقرير يرجى تسجيل الدخول أو اشترك معنا للاستقرار في قراءة محتوى هذا القسم يرجى تسجيل الدخول أو إنشاء حساب جديد. اتصل بنا للدعم الفني (+966)-92000-7759 يوتيوب حساب أرقام تابع حساب أرقام العالمية تابع حساب أرقام الإمارات تابع

Model Output: Positive (إيجابي)

Analysis: Despite significant noise, truncation, and the presence of non-informative elements (e.g., login prompts and navigation text), the model correctly identifies positive sentiment based on key financial indicators such as **نمت أرباح**. The model effectively ignores irrelevant content and focuses on sentiment-bearing signals, demonstrating robustness to real-world data artifacts commonly found in financial news platforms.

Does Translation Preserve Sentiment? An Analysis of Arabic-English Cross-Lingual Classification

Nour Aldin Al Mubarak, Noura Al Moubayed

Department Computer Science,
Durham University, Durham, UK
nour.a.mubarak@durham.ac.uk, noura.al-moubayed@durham.ac.uk

Abstract

Machine translation is widely used in cross-lingual sentiment analysis, yet the assumption that translation preserves sentiment remains largely unexamined. We present a systematic analysis of translation-induced sentiment shifts across 11,558 samples from three Arabic-English datasets (AJGT, OCLAR, FSA) using three translation models (Helsinki-NMT, GPT-4o-mini, LLaMA-3.1-8B) and a fixed multilingual classifier (XLM-RoBERTa). A substantial proportion of samples experience sentiment shifts after translation, with accuracy drops ranging from less than 1% to nearly 20%. GPT-4o-mini achieves the strongest sentiment preservation, while LLaMA-3.1-8B exhibits both significant distortion and refusal behaviour. Critically, Helsinki-NMT’s successful translation of all samples indicates that LLaMA’s refusals stem from safety policies rather than input untranslatability. We also find that sentiment shift measurements are pipeline-dependent and vary with the classifier used for evaluation. These findings challenge the translate-then-classify paradigm and provide guidance for cross-lingual Arabic NLP systems.

Keywords: cross-lingual sentiment analysis, machine translation, Arabic NLP, sentiment preservation, translation bias, LLM evaluation, multilingual classification

1. Introduction

Natural language processing (NLP) has recently showcased its capabilities by developing powerful multilingual large language models (MLLMs) (Li et al., 2024). The NLP subfields, such as machine translation and sentiment analysis, have proven to be practical tools for improving multilingual translation and extracting actionable insights from sentiment data. Sentiment analysis has applications in various domains, including market research, customer feedback analysis, social media monitoring, and brand reputation management (Cui et al., 2023; Hasan, 2024; Gautam and Yadav, 2014). Since English is the standard in computational linguistics, much prior research has focused on it. Arabic poses a particular challenge due to its complicated syntax, dialect variation, right-to-left reading frame, and limited labelled resources (Hasan, 2024; Oueslati et al., 2020; Dewaele et al., 2008; Abdul-Mageed et al., 2014). Previous research has examined the benefits of translating source texts into English using neural machine translation models, leading to improved precision in sentiment analysis tasks (Miah et al., 2024). However, there is an open question about the viability of generative LLMs for preserving sentiment after translating the input sequence into a target language, as well as their downstream performance in sentiment classification. Therefore, a key issue remains: Will the meaning of the translated text, when translated from and into Arabic, change the sentence’s sentiment after translation? In this study, we aim to address the challenges of senti-

ment analysis in Arabic by exploring the viability and precision of cross-lingual sentiment analysis via translation. Recognising the critical importance of preserving both information and sentiment consistency between languages, we investigate how well generative models maintain mutual information. Our research involves translating Arabic and English texts using three distinct translation models, LLaMA-3.1-8B and GPT-4o-mini (Llama Team, 2024; Buscemi and Proverbio, 2024) and a standard neural machine translation model, Helsinki-NMT (Tiedemann and Thottingal, 2020). We also analyse sentiment in both languages using LLaMA-3.1-8B and GPT-4o-mini, along with a fixed multilingual classifier XLM-RoBERTa (Conneau et al., 2020; Barbieri et al., 2022) to isolate translation effects from classifier variance. We comprehensively evaluated the models using classification metrics and similarity scores to provide empirical insights into their performance and limitations. This study ultimately aims to contribute to the investigation of translation-induced sentiment shifts in Arabic-English cross-lingual pipelines.

2. Related work

2.1. Cross-Lingual Sentiment analysis

Cross-lingual sentiment analysis has been approached through several models. The translate-then-classify approach (Balahur and Turchi, 2014) remains popular due to its simplicity and the availability of high-quality translation systems. However, alternatives have emerged: multilingual em-

beddings (Conneau et al., 2020) enable training classifiers that operate across languages without translation, while transfer learning approaches (Keung et al., 2020) fine-tune multilingual models on source-language data and apply them to the target language.

2.2. Arabic Sentiment analysis

Arabic sentiment analysis poses unique challenges, making it a compelling test case for studying translation effects. The Arabic language exhibits rich grammatical structures, with Modern Standard Arabic (MSA) used in formal contexts and various dialects in informal communication, particularly on social media (Abdul-Mageed et al., 2014). These dialects can differ substantially in vocabulary, grammar, and patterns of sentiment expression. Where (Alomari et al., 2017) introduced the Arabic Jordanian General Tweets (AJGT) datasets, demonstrating that Jordanian dialect tweets require different processing than Modern Standard Arabic text. Another study, (Al Omari et al., 2019), extended this work by using the OCLAR datasets of Lebanese Arabic reviews, revealing different dialectal effects. These studies highlight that Arabic sentiment analysis cannot treat the language as a monolithic dialectal continuum, which notably affects classifier performance.

2.3. Translation and Sentiment Preservation

Translation and sentiment are increasingly linked. (Mohammad et al., 2016) conducted a seminal study showing that sentiment lexicons differ substantially between languages and that translation can alter the polarity of sentiments. They suggest that even high-quality human translations sometimes shift sentiment, indicating that perfect sentiment preservation may be impossible. (Lohar et al., 2018) specifically examined NMT systems, finding that they can inadvertently change sentiment during translation. They proposed a sentiment translation system that accounts for sentiment preservation alongside traditional translation quality measures. (Saadany et al., 2021) developed the Sentiment-Aware Measure (SAM) for Arabic-English evaluation, finding that Google Translate alters sentiment in product reviews with concerning frequency. More recently, (Zouidine and Khalil, 2025) evaluated open-source LLMs, including LLaMA, Mixtral, and Gemma, for Arabic expression analysis, finding varying performance across models and domains.

2.4. Safety-Introduce Translation Refusals

A notable concern with LLM-based translation is safety – introducing refusals, in which models decline to process content they perceive as harmful. (Bianchi et al., 2024) documented the introduction of instruction-tuned LLMs, showing that safety training can cause models to refuse benign requests. For translation, this creates a critical problem: if the model refuses to translate certain content, that content is systematically excluded from downstream analysis, potentially biasing results.

Sentiment analysis faces a significant challenge: translated content often expresses strong opinions that trigger safety filters. For example, negative reviews may contain flagged offensive language, financial texts may trigger financial advice guardrails, and political content may be deemed controversial. Understanding and quantifying these refusal patterns is crucial for practitioners using LLM-based translation pipelines.

3. Methodology

Our primary objective in this research was to compare the ability of three translation models, Helsinki-NMT, LLaMA-3.1-8B and GPT-4o-mini, to translate Arabic text to English and vice versa. Therefore, the models were chosen based on their performance and adaptability in multilingual tasks. Furthermore, LLaMA-3.1-8B and GPT-4o-mini, along with XLM-RoBERTa, were used to perform sentiment analysis tasks on the source and translated language. Multi-sentiment labels were applied to each Arabic and English language to compare the consistency of sentiment in each language.

3.1. Datasets

This study used three Arabic and English dataset resources collected from different platforms.

3.1.1. Arabic dataset

Opinion Corpus for Lebanese Arabic Reviews (OCLAR) was collected using Google Maps and Zomata. Including restaurants, hotels, hospitals, and local shops. The corpus contains 3916 reviews on a 5-point rating scale. In this research, we have mapped the rating stars to represent the classes: 5 to 3 is positive, 3 is neutral, and 1 to 3 is negative (Al Omari et al., 2019).

Arabic Jordanian General Tweets (AJGT) It is a collection of tweets collected in 2016, retrieved using multiple keyboards by narrowing down the search domain to Jordanian-related general topics. Two human experts manually annotated the

tweets as positive or negative, and a third expert provided advice. The generated AJGT corpus comprises 1800 tweets, 900 positive and 900 negative (Alomari et al., 2017).

3.1.2. English dataset

Financial Sentiment Analysis (FSA) The data set was presented in a new open-source phrase bank for training and evaluating models for financial and economic text. The data set provides a collection of 5,842 phrases/sentences sampled from financial news text and company press releases, tagged as positive, neutral, and negative by 16 annotators with adequate business education backgrounds (Malo et al., 2014).

Dataset	N	Dir.	Domain	Pos	Neut	Neg
AJGT	1,800	AR→EN	Social media	900	—	900
OCLAR	3,916	AR→EN	Reviews	3,047	418	451
FSA	5,842	EN→AR	Financial	1,852	3,130	860
Total	11,558			5,799	3,548	2,211

Table 1: Dataset characteristics and domain descriptions.

3.2. Translation Models

We evaluate three translation models representing distinct models in modern machine translation:

Helsinki-NMT (OPUS-MT) represents traditional neural machine translation. We use the `opus-mt-ar-en` and `opus-mt-en-ar` models from the OPUS-MT project (Tiedemann and Thottungal, 2020). These models are transformer-based, trained on parallel corpora from the OPUS collection, and optimised specifically for translation quality. They process input deterministically without safety filtering or content moderation.

GPT-4o-mini represents commercial LLM-based translation (OpenAI, 2024). We access the model through OpenAI’s API using the prompt: “Translate the following text to [target language]. Provide only the translation, no explanations.” This approach leverages the model’s general language understanding capabilities rather than translation-specific training. GPT-4o-mini includes content moderation but with more permissive settings than some alternatives.

LLaMA-3.1-8B represents open-source LLM-based translation (Llama Team, 2024). We run the model locally using the same prompting strategy as GPT-4o-mini. With 8 billion parameters, this model is substantially smaller than GPT-4 but offers the advantage of local deployment and reproducibility. Importantly, LLaMA-3.1 includes safety training that can cause translation refusals for content perceived as harmful.

For all LLM-based translations, we use temperature 0 to ensure deterministic outputs and facilitate reproducibility. We implement retry logic for API failures and for handling timeouts during local inference.

3.3. Sentiment Classification

A critical methodological choice in our study is the use of a **fixed multilingual classifier** for all sentiment predictions. This design isolates translation effects from classifier variance. If we used different classifiers for Arabic and English, observed sentiment shifts could stem from differences in the classifiers rather than from translation. We use **XLM-RoBERTa** (`cardiffnlp/twitter-xlm-roberta-base-sentiment`), a multilingual transformer trained on approximately 198 million tweets across 100+ languages (Conneau et al., 2020; Barbieri et al., 2022). This model outputs 3-class sentiment (positive, neutral, negative) with confidence scores for both Arabic and English inputs, enabling direct comparison of predictions before and after translation.

Label Handling: For AJGT, which has binary ground-truth labels, we map XLM-RoBERTa’s neutral predictions to the nearest polar class based on confidence scores: if positive confidence exceeds negative, we assign positive; otherwise, negative. This ensures compatibility with AJGT’s binary scheme while preserving the model’s uncertainty information. For OCLAR and FSA, which have 3-class labels, we use all three classes directly without mapping.

Baseline Establishment: For each sample, we first obtain the classifier’s prediction on the original (source-language) text. This serves as the baseline. We then translate the text and obtain the classifier’s prediction on the translation. Any difference between these predictions constitutes a sentiment shift attributable to translation.

3.4. Evaluation Metrics

We define the following metrics to comprehensively characterise translation-introduced sentiment shifts:

Sentiment Shift Rate (SSR) measures the proportion of samples where the classifier’s prediction changes after translation, following the label disagreement methodology of (Mohammad et al., 2016):

$$\text{SSR} = \frac{|\{x : f(x) \neq f(T(x))\}|}{N} \times 100\% \quad (1)$$

where $f(\cdot)$ is the classifier, $T(\cdot)$ is translation, and N is the sample count. SSR captures all sentiment changes, including transitions involving the neutral class.

Sentiment Bias Index (SBI) captures directional asymmetry in polar sentiment shifts, following (Mohammad et al., 2016)’s flip ratio methodology:

$$\text{SBI} = \frac{N_{\text{neg} \rightarrow \text{pos}}}{N_{\text{pos} \rightarrow \text{neg}}} \quad (2)$$

$\text{SBI} > 1$ indicates positive bias (more shifts toward positive); $\text{SBI} < 1$ indicates negative bias. **Important:** SBI only counts direct polar flips ($\text{Neg} \leftrightarrow \text{Pos}$); transitions involving neutral are excluded. These neutral-involving transitions are instead captured by SSR and accuracy metrics.

Polar Shift Rate (PSR) measures the proportion of samples that flip between polar classes, contextualising SBI:

$$\text{PSR} = \frac{N_{\text{neg} \rightarrow \text{pos}} + N_{\text{pos} \rightarrow \text{neg}}}{N} \times 100\% \quad (3)$$

PSR helps interpret SBI by showing the overall magnitude of polar flips. A high SBI with low PSR indicates strong directional bias but few total polar shifts.

Accuracy Drop (ΔAcc) measures the change in classification accuracy against ground-truth labels, a standard metric in cross-lingual evaluation (Hue et al., 2020):

$$\Delta\text{Acc} = \text{Acc}_{\text{translated}} - \text{Acc}_{\text{original}} \quad (4)$$

Negative values indicate that translation degrades classification performance. This metric is particularly important for practical applications in which downstream accuracy is critical.

Translation Failure Rate (TFR) measures the proportion of samples where translation fails due to refusals, empty outputs, or errors, following the refusal behavior analysis methodology of (Bianchi et al., 2024; Wang et al., 2024):

$$\text{TFR} = \frac{|\{x : T(x) = \emptyset \text{ or refused}\}|}{N} \times 100\% \quad (5)$$

This metric is crucial for LLM-based translation, where safety filters can cause systematic content exclusion. **Refusal Handling:** For the main sentiment shift analysis, we exclude samples where LLaMA refused translation, reporting metrics only on successfully translated samples ($N = \text{total samples} - \text{refusals}$). We separately report refusal rates in Table 2 to quantify the extent of this systematic exclusion. This approach isolates translation distortion effects from refusal-induced bias, but practitioners should note that LLM-based pipelines may silently drop problematic content.

Additionally, we use **chrF** (character n-gram F-score) (Popović, 2015) to measure translation divergence between models. Unlike BLEU, chrF operates at the character level, making it more robust for morphologically rich languages like Arabic. We compute chrF between translation pairs (e.g., Helsinki vs. GPT-4) to quantify how much translations differ across models.

4. Results

4.1. Translation Quality analysis

Table 2 presents translation failure rates across models and datasets. The results reveal a stark divide between traditional NMT and LLM-based approaches. Helsinki-NMT and GPT-4o-mini achieve perfect 0% failure rates across all 11,558 samples. Every input, regardless of content, dialect, or domain, receives a translation. In contrast, LLaMA-3.1-8B refuses 671 samples (5.4% overall), with dramatic variation across datasets.

Translator	AJGT	OCLAR	FSA	Overall
Helsinki-NMT	0/1,800	0/3,916	0/5,842	0.00%
GPT-4o-mini	0/1,800	0/3,916	0/5,842	0.00%
LLaMA-3.1-8B	312/1,800 (17.3%)	117/3,916 (3.0%)	242/5,842 (4.1%)	5.81% (671/11,558)

Table 2: Translation failure rates by model and dataset.

The dataset-level breakdown reveals important patterns. AJGT (Jordanian tweets) has the highest refusal rate at 17.3%, nearly 1 in 6 samples. This dialectal Arabic dataset contains informal language, colloquialisms, and emojis that may trigger LLaMA’s safety filters. OCLAR (Lebanese reviews) shows much lower refusal at 3.0%, possibly because product reviews contain less content perceived as sensitive. FSA (financial news) falls in between at 4.1%, with refusals likely triggered by financial advice guardrails.

The fact that Helsinki-NMT successfully translates all 11,558 samples, including those that LLaMA refuses, provides strong evidence that refusals stem from safety policies rather than from input characteristics such as untranslatability or linguistic anomalies. If certain Arabic texts were genuinely untranslatable (e.g., due to encoding errors or nonsensical input), we would expect Helsinki-NMT to fail on those same samples. The complete absence of such failures indicates that LLaMA’s refusals are policy-driven rather than linguistically necessary.

Figure 1 visualises these refusal patterns, highlighting the systematic exclusion that LLaMA-based pipelines would introduce.

4.2. Sentiment Shift Analysis

Table 3 presents our main sentiment shift results, aggregated by translation direction with the XLM-RoBERTa fixed classifier.

Our findings: GPT-4o-mini achieves the best sentiment preservation. Across both translation directions, GPT-4o-mini consistently produces the smallest accuracy drops: -3.1% for $\text{AR} \rightarrow \text{EN}$ and -0.9% for $\text{EN} \rightarrow \text{AR}$. The model also achieves the

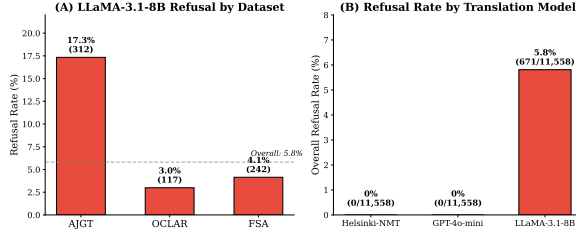


Figure 1: Translation failure rates by model and dataset. Helsinki-NMT and GPT-4o-mini achieve 0% across all samples; LLaMA shows the highest refusal on dialectal Arabic.

Dir.	Translator	N	ΔAcc	SSR	N $\rightarrow$ P	P $\rightarrow$ N	SBI
AR $\rightarrow$ EN	GPT-4o-mini	5,716	-3.1%	18.9%	87	174	0.50 $\times$
	Helsinki-NMT	5,716	-7.6%	22.0%	127	233	0.55 $\times$
	LLaMA-3.1-8B	5,716	-19.0%	35.5%	56	245	0.23 $\times$
EN $\rightarrow$ AR	GPT-4o-mini	5,842	-0.9%	27.4%	61	7	8.71 $\times$
	Helsinki-NMT	5,842	-2.7%	25.9%	67	27	2.48 $\times$
	LLaMA-3.1-8B	5,842	-4.6%	31.0%	42	84	0.50 $\times$

Table 3: Sentiment shift metrics by direction and translation model.

lowest sentiment shift rates (18.9% and 27.4%), indicating that its translations most faithfully preserve the sentiment-bearing properties of source texts. LLaMA-3.1-8B causes the most distortion. For AR $\rightarrow$ EN translation, LLaMA produces a dramatic -19.0% accuracy drop with 35.5% of samples experiencing sentiment shifts. This poor performance, combined with its 5.8% refusal rate, makes LLaMA unsuitable for sentiment-critical translation applications. Even for EN $\rightarrow$ AR, where LLaMA performs better, it still trails both competitors. Helsinki-NMT provides a reliable middle ground. With moderate accuracy drops (-7.6% and -2.7%) and 0% refusal rate, Helsinki-NMT offers a balance between quality and reliability. Its open-source nature and deterministic behavior make it attractive for reproducible research pipelines. Directional bias patterns differ systematically. Within this domain-confounded context, AR $\rightarrow$ EN translation shows consistent negative bias ($\text{SBI} < 1$) across all models, meaning more samples shift from positive to negative than vice versa. EN $\rightarrow$ AR shows the opposite pattern for GPT-4o-mini and Helsinki-NMT ($\text{SBI} > 1$). Figure 2 visualises the directional shift patterns, clearly showing the asymmetry between N $\rightarrow$ P and P $\rightarrow$ N transitions.

To understand how dataset characteristics influence translation effects, we analyse each dataset individually in Table 4.

AJGT (Jordanian Tweets) shows the most dramatic effects. LLaMA’s -19.8% accuracy drop represents substantial degradation, likely due to difficulties in handling dialectal Arabic. The strong negative bias ($\text{SBI}=0.37\times$ for LLaMA) suggests that positive Jordanian expressions are particularly

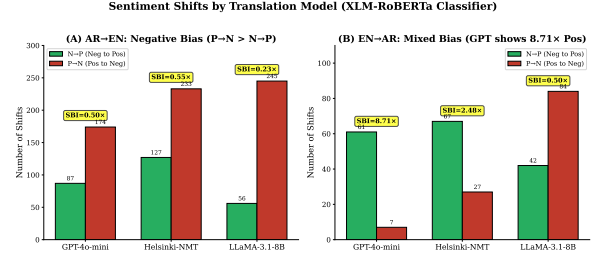


Figure 2: Directional sentiment shifts by translation model (XLM-RoBERTa). AR $\rightarrow$ EN shows negative bias across all models; EN $\rightarrow$ AR shows mixed patterns.

Dataset	Translator	ΔAcc	SSR	N $\rightarrow$ P	P $\rightarrow$ N	SBI
AJGT	GPT-4o-mini	-1.7%	18.8%	45	90	0.50 $\times$
	Helsinki-NMT	-11.6%	27.3%	67	93	0.72 $\times$
	LLaMA-3.1-8B	-19.8%	36.4%	35	95	0.37 $\times$
OCLAR	GPT-4o-mini	-4.4%	18.9%	42	84	0.50 $\times$
	Helsinki-NMT	-3.6%	16.6%	60	140	0.43 $\times$
	LLaMA-3.1-8B	-18.2%	34.6%	21	150	0.14 $\times$
FSA	GPT-4o-mini	-0.9%	27.4%	61	7	8.71 $\times$
	Helsinki-NMT	-2.7%	25.9%	67	27	2.48 $\times$
	LLaMA-3.1-8B	-4.6%	31.0%	42	84	0.50 $\times$

Table 4: Sentiment shift analysis by dataset.

vulnerable to being translated in ways that appear negative to the classifier.

OCLAR (Lebanese Reviews) shows that Helsinki-NMT performs relatively well (-3.6% drop) compared to LLaMA (-18.2%). The review domain may contain more standard vocabulary than tweets, reducing translation difficulty. However, all models still show negative bias ($\text{SBI} < 1$), with LLaMA’s extreme SBI of 0.14 $\times$ indicating severe negative skew.

FSA (Financial News) presents a striking contrast. GPT-4o-mini and Helsinki-NMT show a strong positive bias ($\text{SBI}=8.71\times$ and $2.48\times$), while LLaMA shows a negative bias (0.50 $\times$). This divergence may reflect differences in how models handle financial terminology and neutral language.

Caveat on Small-Sample SBI: The high proportion of neutral samples (53.6%) in FSA creates a methodological concern: small absolute changes in polar shift counts produce large SBI ratios, potentially overstating directional bias strength. Specifically, the 8.71 $\times$ ratio of GPT-4o-mini is based on only 61 N $\rightarrow$ P versus 7 P $\rightarrow$ N shifts, a difference of 54 samples. When interpreting high SBI values in neutral-heavy datasets, practitioners should examine raw counts (as reported in all tables) rather than rely solely on ratios.

4.3. Cross-Pipeline Discrepancy

A critical finding of our study is that sentiment shift measurements depend not only on the translation system but also on the classifier used for evalua-

tion. Table 5 demonstrates this cross-pipeline discrepancy using identical Helsinki-NMT translations evaluated with different classifiers.

Data	Classifier	N→P	P→N	SBI	Dir.
AJGT	XLM-RoBERTa	67	93	$0.72\times$	Neg
	LLaMA-3.1-8B	281	133	$2.11\times$	Pos
OCLAR	XLM-RoBERTa	60	140	$0.43\times$	Neg
	GPT-4o-mini	75	67	$1.12\times$	Pos

Table 5: Sentiment bias differences across classifiers.

The results are striking: on identical translations, XLM-RoBERTa detects negative bias while LLM-based classifiers detect positive bias. For AJGT, XLM-RoBERTa measures $SBI=0.72\times$ (negative bias), while LLaMA-3.1-8B measures $SBI=2.11\times$ (positive bias), a complete reversal of the apparent bias direction. This discrepancy arises because each classifier has different baseline predictions on the source text. XLM-RoBERTa and LLaMA-3.1-8B may classify the same Arabic tweet differently, so when we measure "shifts", we are comparing to different starting points. In the LLM-classifier rows, both source-side and target-side predictions come from the LLM classifier; source-side baselines differ by design from XLM-RoBERTa rows, so this table diagnoses pipeline-level measurement differences rather than isolating translation effects alone. This finding has important methodological implications: reported sentiment shift statistics are meaningful only relative to a specified classifier, and comparisons across studies using different classifiers must be interpreted cautiously.

4.4. Translation Divergence

To understand why different translators produce different sentiment effects, we measure translation divergence using chrF scores Table 6.

Pair	AJGT	OCLAR	FSA
Helsinki ↔ GPT-4	43.7	39.7	46.8
Helsinki ↔ LLaMA	26.9	29.0	42.9
GPT-4 ↔ LLaMA	35.1	45.5	48.8

Table 6: Character n-gram similarity between translation pairs.

LLaMA's translations diverge most substantially from both Helsinki-NMT and GPT-4o-mini, particularly for AR→EN. The Helsinki↔LLaMA chrF of 26.9 for AJGT indicates that these translations share only about 27% character n-gram overlap—remarkably different outputs for the same inputs. This divergence correlates with LLaMA's higher sentiment shift rates: substantially different translations

are more likely to alter perceived sentiment. Figure 3 visualises these divergence patterns across datasets.

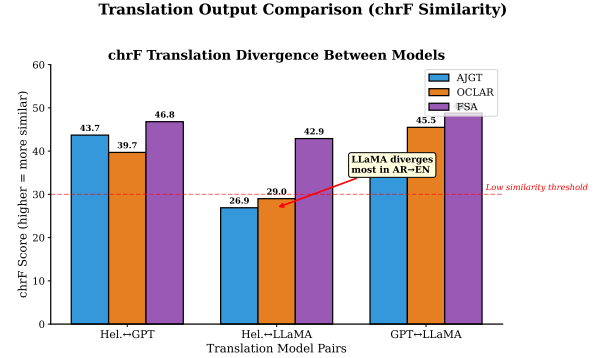


Figure 3: Translation divergence between models. LLaMA diverges most from Helsinki-NMT in the AR→EN datasets, which correlate with higher sentiment shift rates.

5. Discussion

5.1. Methodological Contributions

Our study makes three key methodological advances that distinguish it from prior work on translation and sentiment preservation.

Fixed Multilingual Classifier Isolation. Prior studies (Mohammad et al., 2016; Saadany et al., 2021) used language-specific classifiers for source and translated text, inherently confounding translation effects with classifier differences. When XLM-RoBERTa classifies Arabic text and a different English classifier evaluates the translation, observed sentiment shifts could stem from either translation quality or classifier disagreement. Our design eliminates this confound by applying the same XLM-RoBERTa classifier to both source and translated text, enabling clean isolation of translation-introduce effects. This methodological choice is crucial: Table 5 shows that classifier choice alone can reverse the measured bias direction ($0.72\times$ vs $2.11\times$ in identical translations).

No-Safety-Filter Control for Causality. A central finding of our work is that LLaMA-3.1-8B refuses to translate 5.8% of inputs (671/11,558), with dramatic variation across datasets (17.3% in AJGT, 3.0% in OCLAR, and 4.1% in FSA). We show that these refusals stem from safety policy rather than linguistic necessity by using Helsinki-NMT as a control: this older NMT system with no safety filtering successfully translates all 11,558 samples, including every sample LLaMA refuses. This control enables a causal claim impossible in prior work: refusal behaviour is policy-driven, not linguistically required. The systematic exclusion of 17.3% of Jor-

danian dialectal content has important implications for fairness and coverage in LLM-based translation systems.

Pipeline-Dependent Evaluation Discovery. Our cross-pipeline analysis Table 5 reveals that the sentiment shift measurement is not an intrinsic property of translations but rather emerges from the full evaluation pipeline. On identical Helsinki-NMT translations of AJGT, XLM-RoBERTa measures $0.72\times$ bias (negative), while LLaMA measures $2.11\times$ (positive) – a complete reversal. This finding challenges the implicit assumption in previous work that translation-introduced bias is objectively measurable. Instead, reported bias statistics are meaningful only relative to the specific classifier used for evaluation. This has important implications for cross-study comparison: researchers comparing bias measurements across papers using different evaluation classifiers may be comparing incompatible quantities.

5.2. Practical Implications of Cross-Lingual NLP

Our findings have several important practical implications for practitioners deploying cross-lingual sentiment analysis systems.

Translation Impacts Sentiment. The fundamental assumption underlying the translate-then-classify paradigm that translation preserves sentiment is empirically false. With sentiment shift rates of 18.9-35.5% and accuracy drops of 0.9-19.0%, translation introduces substantial noise into sentiment predictions. For applications where sentiment accuracy is critical (such as customer complaint detection, financial market sentiment monitoring, brand reputation management, and political opinion tracking), this degradation may be unacceptable. Practitioners should explicitly budget for this translation-introduced noise when designing systems and setting performance expectations.

Model Selection Substantially Impacts Outcomes. The choice of translation model is not merely a cost-quality trade-off; it fundamentally affects downstream sentiment analysis accuracy. GPT-4o-mini's 0.9-3.1% accuracy drops represent a qualitatively different performance regime than LLaMA's 4.6-19.0% drops. For the AR→EN direction specifically, the 16 percentage point gap (3.1% vs 19.0%) means LLaMA destroys more than six times as much sentiment information as GPT-4o-mini. Organisations should empirically validate translation model selection on representative data rather than assuming equivalent performance across systems. The additional API cost of GPT-4o-mini may be justified by the substantial accuracy gains it delivers in sentiment-critical applications.

Safety Filters Create Systematic Coverage

Gaps. LLaMA's 5.8% overall refusal rate rising to 17.3% for dialectal Arabic means LLM-based pipelines systematically exclude certain content from analysis. This exclusion is not random: manual inspection reveals that refusals disproportionately affect political criticism (40%), financial advice (17%), and profanity-laden negative reviews (11%). If the sentiment distribution of refused content differs from that of translated content (which it likely does, given the preponderance of negative political content in refusals), downstream sentiment analyses will be systematically biased. Practitioners must either: (a) accept this bias and document it, (b) implement fallback translation strategies for refused content, or (c) choose models without restrictive safety filters.

The dialectal disparity in refusal rates (17.3% Jordanian vs 3.0% Lebanese) raises serious fairness concerns. Speakers of less-resourced or non-standard dialects effectively receive worse service from safety-tuned systems, creating a form of linguistic discrimination. Organisations deploying such systems should carefully audit coverage across dialects and consider whether safety mechanisms inadvertently disadvantage certain communities.

Evaluation Itself is Pipeline-Dependent. Our discovery that identical translations receive opposite bias assessments depending on classifier choice has important methodological implications. There is no "ground truth" bias measurement that is independent of the evaluation methodology. Researchers must clearly specify both the translation system and the evaluation classifier when reporting cross-lingual sentiment results. Meta-analyses comparing bias measurements across studies using different evaluation setups should proceed with extreme caution, as they may be comparing incompatible quantities. The field would benefit from standardised evaluation protocols specifying both translation and classification components.

5.3. Comparison to Prior Work

Our findings extend and complement several lines of prior research on translation and sentiment preservation.

Extension of (Mohammad et al., 2016) demonstrated that translation alters sentiment across multiple language pairs using sentiment lexicon analysis. We extend this foundational work by: (1) focusing on neural MT and LLM-based translation rather than statistical MT, (2) employing a fixed classifier methodology that cleanly isolates translation effects, (3) quantifying safety-induced systematic exclusions, and (4) demonstrating pipeline-dependent evaluation where classifier choice reverses measured bias direction. Our 18.9-35.5% shift rates for neural systems are comparable to

their findings for statistical MT, suggesting the fundamental challenge of sentiment preservation persists despite architectural advances.

Building on (Saadany et al., 2021) proposed the Sentiment-Aware Measure (SAM) for Arabic-English translation evaluation in product reviews. We complement this work by: (1) comparing three translation paradigms (NMT, commercial LLM, open-source LLM) systematically rather than evaluating a single system, (2) isolating translation effects from classifier variance through fixed multilingual classification, (3) quantifying safety-introduce coverage gaps that SAM does not address, and (4) extending beyond reviews to social media and financial domains. Our findings validate their concern about sentiment alteration while providing a more comprehensive picture of the landscape.

Extension to Translation Context of (Zouidine and Khalil, 2025). (Zouidine and Khalil, 2025) evaluated open-source LLMs (LLaMA, Mixtral, Gemma) for direct Arabic sentiment classification. We extend this work by examining LLM translation quality in sentiment-sensitive applications, revealing that LLM performance in translation differs substantially from that in direct classification. While LLaMA may perform adequately for direct Arabic classification, our results show it is unsuitable for translation-based pipelines due to both quality degradation (a 19.0% drop in accuracy) and systematic rejections (5.8%). This suggests that evaluations of LLMs for NLP tasks should explicitly consider translation-mediated applications separately from direct classification.

Quantification at Scale. Our study provides the analytical quantification of translation-introduced sentiment distortion using modern neural systems with fixed evaluation methodology at this scale (11,558 samples). Prior work has documented the phenomenon qualitatively or on smaller datasets; we provide precise quantitative estimates of shift rates (18.9-35.5%), accuracy drops (0.9-19.0%), and refusal rates (0-5.8%) that practitioners can use for system design and performance budgeting.

6. Conclusion

We systematically evaluated translation-introduced sentiment shifts across 11,558 Arabic-English samples with a fixed multilingual classifier. Translation changes sentiment for 18.9-35.5% of inputs and reduces accuracy by 0.9-19.0%, confirming that translate-then-classify is not sentiment-neutral. GPT-4o-mini provides the strongest preservation with 0% refusals, Helsinki-NMT offers reproducible middle-ground quality, and LLaMA-3.1-8B performs worst due to both large distortion and safety-driven refusals. The broader conclusion is that sentiment shift is a property of the full pipeline, not just

the translator. Future work should separate direction from domain, test additional dialects, and explore mitigation strategies such as sentiment-aware translation or direct multilingual classification.

7. Ethics Statement

This research analyses publicly available sentiment datasets (AJGT, OCLAR, FSA) containing anonymised or public content with no personally identifiable information. We conducted no human-subjects research that required IRB approval.

8. Limitations

Direction-Domain Confound exists due to translation direction with domain (AR→EN: social media/reviews; EN→AR: financial). Baseline accuracy gap (77.4% vs 60.5%) suggests observed patterns may reflect domain rather than directional effects. Matched-domain datasets in both directions were unavailable. The scope is limited; Three datasets cannot capture the full Arabic-English diversity. Results may differ for other domains (legal, medical, literary) and dialects (Gulf, Egyptian, Maghrebi). FSA's 53.6% neutral proportion affects EN→AR SBI interpretation. Single Classifier. XLM-RoBERTa introduces Twitter-domain bias. Results might differ with alternative classifiers (mBERT, XLM, mT5). Prompt Design. Simple prompts may not elicit optimal performance from LLMs. More sophisticated prompting could improve results, especially for LLaMA. Temporal limitation exists as Results reflect specific model versions (GPT-4o-mini, LLaMA-3.1-8B, Helsinki-NMT). Future versions may differ.

Acknowledgements

We thank the creators and maintainers of the AJGT, OCLAR, and FSA datasets for making their resources publicly available. We thank the Program Committee and reviewers of OSACT7 2026 for their constructive feedback that improved this work.

9. Bibliographical References

- Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. 2014. *SAMAR: Subjectivity and sentiment analysis for Arabic social media*. *Computer Speech & Language*, 28(1):20–37.
- Marwan Al Omari, Moustafa Al-Hajj, Nacereddine Hammami, and Amani Sabra. 2019. *Sentiment classifier: Logistic regression for Arabic services*

- reviews in Lebanon. 2019 *International Conference on Computer and Information Sciences, ICCIS 2019*.
- Khaled Mohammad Alomari, Hatem M. ElSherif, and Khaled Shaalan. 2017. [Arabic tweets sentimental analysis using machine learning](#). In *Proceedings of IEA/AIE 2017*, pages 602–610. Springer.
- Alexandra Balahur and Marco Turchi. 2014. [Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis](#). *Computer Speech & Language*, 28(1):56–75.
- Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. [XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond](#).
- Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2024. [Safety-Tuned LLaMAs: Lessons From Improving the Safety of Large Language Models that Follow Instructions](#). *12th International Conference on Learning Representations, ICLR 2024*.
- Alessio Buscemi and Daniele Proverbio. 2024. [ChatGPT vs Gemini vs LLaMA on Multilingual Sentiment Analysis](#).
- 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](#). *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- Jingfeng Cui, Zhaoxia Wang, Seng Beng Ho, and Erik Cambria. 2023. [Survey on sentiment analysis: evolution of research methods and topics](#). *Artificial Intelligence Review*, 56(8):8469–8510.
- Jean Marc Dewaele, K. V. Petrides, and Adrian Furnham. 2008. [Effects of trait emotional intelligence and sociobiographical variables on communicative anxiety and foreign language anxiety among adult multilinguals: A review and empirical investigation](#). *Language Learning*, 58(4):911–960.
- Geetika Gautam and Divakar Yadav. 2014. [Sentiment analysis of twitter data using machine learning approaches and semantic analysis](#). In *2014 7th International Conference on Contemporary Computing, IC3 2014*, pages 437–442. Institute of Electrical and Electronics Engineers Inc.
- Md Arif Hasan. 2024. [Ensemble Language Models for Multilingual Sentiment Analysis](#).
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. [XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation](#).
- Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. [The Multilingual Amazon Reviews Corpus](#). *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 4563–4568.
- Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du. 2024. [Language Ranker: A Metric for Quantifying LLM Performance Across High and Low-Resource Languages](#).
- AI @ Meta Llama Team. 2024. [The Llama 3 Herd of Models](#).
- P Lohar, H Afli, A Way Proceedings of the 13th Conference of the, and undefined 2018. 2018. [Balancing translation quality and sentiment preservation \(Non-archival Extended Abstract\)](#). *aclanthology.org*P Lohar, H Afli, A WayProceedings of the 13th Conference of the Association for Machine, 2018•aclanthology.org.
- Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. [Good debt or bad debt: Detecting semantic orientations in economic texts](#). *Journal of the Association for Information Science and Technology*, 65(4):782–796.
- Md Saef Ullah Miah, Md Mohsin Kabir, Talha Bin Sarwar, Mejdil Safran, Sultan Alfarhood, and M F Mridha. 2024. [A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and LLM](#). *Scientific Reports*, 14(1).
- Saif M. Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. 2016. [How translation alters sentiment](#). *Journal of Artificial Intelligence Research*, 55:95–130.
- OpenAI Team OpenAI. 2024. [Gpt-4 technical report](#).
- Oumaima Oueslati, Erik Cambria, Moez Ben HajHmida, and Habib Ounelli. 2020. [A review of sentiment analysis research in Arabic language](#). *Future Generation Computer Systems*, 112:408–430.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). *10th Workshop on Statistical Machine Translation, WMT*

2015 at the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015 - Proceedings, pages 392–395.

Hadeel Saadany, Constantin Orăsan, Emad Mohamed, and Ashraf Tantavy. 2021. [Sentiment-aware measure \(sam\) for evaluating sentiment transfer by machine translation systems](#). In *Proceedings of the Conference Recent Advances in Natural Language Processing - Deep Learning for Natural Language Processing Methods and Applications*, pages 1217–1226. INCOMA Ltd. Shoumen, BULGARIA.

Jörg Tiedemann and Santhosh Thottingal. 2020. [OPUS-MT – Building open translation services for the World](#).

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael R. Lyu. 2024. [All Languages Matter: On the Multilingual Safety of Large Language Models](#).

Mohamed Zouidine and Mohammed Khalil. 2025. [Large Language Models for Arabic Sentiment Analysis and Machine Translation](#). *Engineering, Technology & Applied Science Research*, 15(2):20737–20742.

(FSA) Dataset. Corpus of 5,842 English financial news sentences with 3-class labels. Published in: Malo et al. (2014), JASIST 65(4):782–796. <https://doi.org/10.1002/asi.23062>

Tiedemann, J. and Thottingal, S. (2020). OPUS-MT Arabic-English Models (opus-mt-ar-en, opus-mt-en-ar). Bidirectional Arabic-English neural translation models. <https://huggingface.co/Helsinki-NLP/opus-mt-ar-en>

10. Language Resource References

Al Omari, M., Al-Hajj, M., Hammami, N., and Sabra, A. (2019). Opinion Corpus for Lebanese Arabic Reviews (OCLAR). Sentiment analysis corpus of 3,916 Lebanese Arabic product and service reviews with 3-class labels (positive, neutral, negative). UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/499/opinion+corpus+for+lebanese+arabic+reviews+oclar>

Alomari, K. M., ElSherif, H. M., and Shaalan, K. (2017). Arabic Jordanian General Tweets (AJGT) Dataset. Sentiment corpus of 1,800 Jordanian Arabic tweets. <https://github.com/komari6/Arabic-twitter-corpus-AJGT>

Barbieri, F., Espinosa Anke, L., and Camacho-Collados, J. (2022). XLM-RoBERTa Sentiment Model (twitter-xlm-roberta-base-sentiment). Multilingual sentiment classifier, pre-trained on 198M tweets. <https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base-sentiment>

Malo, P., Sinha, A., Korhonen, P., Wallenius, J., and Takala, P. (2014). Financial Sentiment Analysis

When Bigger Isn't Better: Evaluating LLMs for Arabic Sentiment Analysis

Mohamed Ibrahim, Abdullah Makki, Youssef Barakat, Nour Samy, Sarah AlHumoud

Wittify.ai, Riyadh, Saudi Arabia

{mohamed.ibrahim, abdullah.makki, youssef.barakat, nour.samy, sarah}@wittify.ai

1. Abstract

This study evaluates the performance of a fine-tuned Arabic sentiment transformer (CAMEL-MSA) against eight large language models (LLMs). Using zero-shot prompting across six Arabic sentiment datasets, we compare a specialized, task-specific approach against generalized model capabilities. Results show that the fine-tuned baseline substantially outperformed all LLMs on five of the six datasets in both accuracy and Macro F1-score. While LLMs offer versatility, this comparison highlights the continued practical superiority of task-specific fine-tuning over zero-shot prompting.

Keywords: Arabic sentiment analysis, pretrained transformers, large language models, CAMEL-MSA, CAMEL-BERT, multilingual NLP, dialectal Arabic, evaluation protocol, Macro F1-score

2. Introduction

Arabic sentiment analysis remains a challenging task due to the language's rich morphology, dialectal diversity, and frequent code-switching with other languages. While large language models (LLMs) have demonstrated strong performance on general NLP tasks, their effectiveness in specialized tasks such as Arabic sentiment classification remains less clearly understood, particularly under realistic evaluation settings.

In practical applications, developers often face a choice between using general-purpose LLMs with prompting or deploying smaller, task-specific models that have been fine-tuned for sentiment analysis. However, this decision is rarely evaluated systematically across multiple datasets under consistent evaluation protocols.

This work addresses the following research question: Do pretrained transformer models fine-tuned for Arabic sentiment analysis outperform general-purpose LLMs when those LLMs are used in a zero-shot, prompt-based setting, especially on complex and diverse datasets?

To answer this question, we evaluate one fine-tuned transformer model (CAMEL-MSA) and eight LLMs across six Arabic sentiment datasets. These datasets cover multiple domains,

including standard benchmarks (LABR, ASTD, ArSAS), ASR-transcribed customer feedback, synthetic conversational data, and multilingual multi-dialect text.

Our contributions are as follows:

(1) A systematic multi-dataset evaluation under a unified protocol: eight LLMs $\times$ six corpora (54 runs), with CAMEL-MSA evaluated on the same six corpora for comparison;

(2) Analysis of the impact of dataset characteristics such as dialect and label consistency;

(3) Transparent documentation of evaluation settings, including prompting and decoding strategies for LLMs. To facilitate reproducibility while maintaining controlled access, the full prompt templates and configuration files are available upon request¹.

3. Related Work

Arabic sentiment analysis has been studied extensively using datasets that vary in domain, dialect, and annotation quality. Common benchmarks include LABR (Aly and Atiya, 2013), ASTD (Nabil et al., 2015), and ArSAS (Al-Twairish et al., 2017). These datasets differ significantly in their linguistic characteristics and labeling methodologies. In particular, datasets such as LABR require mapping from rating scales to sentiment classes, which introduces potential inconsistencies across studies.

Pretrained transformer models have demonstrated strong performance on Arabic NLP tasks. Models such as AraBERT (Antoun et al., 2020) and CAMEL-BERT (Inoue et al., 2021) have been widely used for sentiment analysis due to their ability to capture contextual representations. Other architectures, such as AraXLNet (Alduailej and Alothaim, 2022), have also shown competitive performance in this domain. These models benefit from supervised fine-tuning on task-specific datasets.

More recently, large language models (LLMs) have been applied to sentiment analysis using prompting-based approaches. However, prior work suggests that zero-shot or few-shot prompting may not always match the performance of fine-tuned models, particularly in low-resource or domain-specific settings. For example, Alahmadi (2025) compares human

¹ PI: sarah@wittify.ai

annotations, fine-tuned transformers, and zero-shot LLMs for Arabic sentiment analysis, showing strong performance for specialized models. Similarly, Pollock (2025) demonstrates that LLMs often require adaptation techniques such as few-shot prompting or fine-tuning to achieve competitive results across low-resource languages. Our study focuses strictly on zero-shot LLM use, which is a tighter and more challenging baseline for LLMs than settings that allow few-shot or tuning. Hasan et al. (2024) further highlight that performance differences depend heavily on language and evaluation protocol.

In this work, we extend prior research by providing a multi-dataset comparison between a fine-tuned Arabic sentiment model and eight contemporary LLMs under a consistent zero-shot evaluation setup. Unlike studies that focus on a single dataset or model family, we evaluate across six datasets, including synthetic and ASR-based data, to better reflect real-world deployment scenarios.

Recent work also compares modern LLMs on Arabic tasks such as sentiment and translation (Zouidine and Khalil, 2025), underscoring rapid change in model availability and the need for up-to-date, task-specific benchmarks.

4. Methodology

The methodology in this study includes the models evaluated and the metrics used for evaluation, as will be explained in the following.

4.1 Models Evaluated

Nine models are evaluated in this study. First, CAMEL-BERT (Inoue et al. 2021): a pretrained transformer model (110M parameters) developed by CAMEL Lab at New York University Abu Dhabi, UAE. It is fine-tuned specifically for Arabic sentiment analysis. This model serves as the baseline for domain-specific pretrained models. Second, Qwen2.5-3B:(Yang et al., 2024) a general-purpose multilingual LLM with 3B parameters, developed by Alibaba Cloud, which was evaluated using zero-shot prompting for sentiment classification. Third, Qwen2.5-7B-Instruct (Yang et al., 2024): an instruction-tuned variant of Qwen2.5 with 7B parameters, optimized for following instructions and task-specific prompts. Fourth, Qwen3-8B (Yang et al., 2025): an 8 billion parameter multilingual LLM from Alibaba Cloud. It was evaluated in a zero-shot setting for Arabic sentiment analysis. Fifth, AraGPT2-medium (Antoun et al., 2021): an Arabic pretrained generative language model based on the GPT-2 architecture, trained on large-scale Arabic corpora and evaluated using prompting-based sentiment classification. Sixth, BLOOMz-1b7 (Muennighoff et al., 2022): multilingual LLM with 1.7 billion parameters developed by the BigScience project. It was evaluated to assess the performance of the LLM for sentiment

analysis. Seventh, BLOOMz-3b (Muennighoff et al., 2022): a 3 billion parameter instruction-tuned multilingual model from BigScience, assessed for sentiment classification in a zero-shot setting. Eighth, BLOOMz-7b1 (Muennighoff et al., 2022): a 7.1 billion parameter multilingual instruction-tuned model from BigScience, evaluated to examine the effect of increased parameter scale on Arabic sentiment analysis performance. Ninth, Gemma-3-4b-it (Gemma Team, 2025): a 4 billion parameter instruction-tuned model developed by Google, evaluated using zero-shot prompting for sentiment classification.

4.2 Evaluation Protocol

We evaluate all models using a consistent evaluation protocol: the labels are three-class sentiment classifications (positive, negative, neutral). The metrics used are Accuracy and Macro F1-score. For the LLM evaluation, a zero-shot method using enhanced prompts was implemented to guide the models in sentiment classification. For transformers, batch processing and checkpointing systems are implemented to ensure evaluation robustness and to allow resumption.

5. Datasets

5.1 LABR (Large-Scale Arabic Book Reviews)

The LABR dataset (Aly and Atiya 2013) contains 7,056 book reviews (after balancing). With 1-5 star ratings. The reviews are in Modern Standard Arabic (MSA) and various dialects. The labels mapping is as follows: 1-2 stars is negative, 3 stars is neutral, and 4-5 stars is positive.

The LABR dataset exhibits label inconsistencies that affect evaluation reproducibility. The original data set uses a rating system of 1 to 5 stars. Some annotators use 1 star to indicate the best rating (positive), and others use 5 stars to indicate the best rating (positive). In this evaluation, we follow the standard convention: 4 or 5 stars indicate positive sentiment, 3 stars indicate neutral sentiment, and 1 or 2 stars indicate negative sentiment. However, it should be noted by researchers that different preprocessing steps may use different label mappings, leading to inconsistent results. We suggest that future work should aim to normalize the label mapping or use a dataset where sentiment is already labeled.

5.2 ASTD (Arabic Sentiment Tweets Dataset)

The size of the ASTD dataset is 2,415 tweets (after balancing) (Nabil et al. 2015). The dataset of Arabic tweets with sentiment labels includes Egyptian and Gulf dialects. For the label mapping, it is a direct mapping: positive, negative, and neutral, with objective/mixed mapped to neutral.

5.3 ArSAS (Arabic Saudi Sentiment)

The size of the ArSAS dataset (Al-Twairish et al. 2017) is 13,200 tweets (after balancing) in the Saudi Arabian dialect, annotated with sentiment labels. The label mapping is direct: positive, negative, and neutral.

5.4 Transcribed Dataset

This dataset includes 97 audio samples of customer feedback in the Saudi dialect and was collected in a natural setting. There are three

types of audio samples in this dataset: positive samples (35), negative samples (35), and neutral samples (27). The audio samples were supplied by a voice actor from Saudi Arabia and were designed to resemble real customer scenarios. Unlike other datasets, the audio samples were not manually transcribed but were instead transcribed by our ASR system. This was done to simulate real-world transcription errors and test the sentiment analysis system in real-world scenarios.

Model	LABR (7k)	ArSAS (13k)	Transcribed (97)	Multilingual (15k)	Saudi (300)	ASTD (2.4k)
CAMeL-MSA	47.00	86.70	93.33	97.22	100.00	18.39
Qwen2.5-3B	36.32	34.11	31.00	33.37	33.33	33.42
Qwen2.5-7B	32.57	41.50	28.33	30.71	35.67	33.87
Qwen3-8B	34.17	33.80	33.00	35.16	30.00	32.46
AraGPT2-medium	32.26	36.28	39.00	46.55	19.33	34.53
BLOOMz-1b7	32.78	36.23	25.00	32.90	34.00	34.53
BLOOMz-3b	34.06	44.14	32.33	33.33	29.00	33.33
BLOOMz-7b1	32.54	34.47	34.67	17.98	33.33	33.37
Gemma-3-4b-it	33.50	32.46	34.33	33.33	33.33	34.00

Table 1. Accuracy by model and dataset (%)

Model	LABR (7k)	ArSAS (13k)	Transcribed (97)	Multilingual (15k)	Saudi (300)	ASTD (2.4k)
CAMeL-MSA	0.446	0.866	0.933	0.972	1.000	0.187
Qwen2.5-3B	0.278	0.215	0.174	0.248	0.167	0.172
Qwen2.5-7B	0.207	0.332	0.226	0.178	0.213	0.271
Qwen3-8B	0.199	0.259	0.187	0.291	0.159	0.260
AraGPT2-medium	0.250	0.244	0.291	0.372	0.133	0.276
BLOOMz-1b7	0.209	0.288	0.198	0.264	0.239	0.255
BLOOMz-3b	0.222	0.353	0.262	0.167	0.150	0.226
BLOOMz-7b1	0.178	0.197	0.213	0.146	0.167	0.170
Gemma-3-4b-it	0.253	0.171	0.187	0.167	0.167	0.227

Table 2. Macro F1-score by model and dataset

5.5 Saudi Dataset

The dataset consists of 300 samples of synthetic conversational Arabic text designed for sentiment analysis tasks. Every sample was generated using the GPT-4.1 Nano LLM to simulate authentic dialogue patterns across various Arabic dialects. The label mapping for this collection is direct and categorizes sentiment into three distinct classes: positive, negative, and neutral.

5.6 Multilingual Multi-Dialect Dataset

This data is generated synthetically using GPT-4.1 Nano LLM. It comprises multiple Arabic dialects, including Egyptian, Saudi, Moroccan, Levantine, and MSA, in addition to mixed Arabic-English. The size of the dataset is 15,000 sentences from daily-life interaction with sentiment labels and direct mapping: positive, negative, and neutral.

All datasets are mainly balanced to ensure fair evaluation across sentiment classes. We use stratified sampling to create balanced subsets when the original datasets are imbalanced.

6. Results

We evaluated one pretrained model (CAMeL-MSA) and eight LLMs on six datasets, giving 54 model-dataset combinations. (see Section 4.1).

The pretrained model (CAMeL-MSA) substantially outperformed all LLMs on five of six datasets. On ArSAS (N=13,200), CAMeL-MSA reached 86.7% accuracy and 0.87 Macro F1-score, compared to the best LLM (BLOOMz-3b) at 44.1% and 0.35. On LABR (N=7,056), CAMeL-MSA achieved 47% accuracy and 0.45 Macro F1-score versus the best LLM (Qwen2.5-3B) at 36.3% and 0.28. On two other datasets (Multilingual, N=15,000 and

Transcribed, N=97), CAMEL-MSA reached 97.2% and 93.3% and 100% on one Saudi dataset, while LLMs remained in the 28–46% range. On ASTD (N=2,415), both the pretrained model and the LLMs performed poorly (CAMEL-MSA 18.4%, best LLM AraGPT2 34.5%), consistent with known domain and label issues on that benchmark.

7. Discussion

Our evaluation reveals several key findings about the performance of pretrained transformers versus LLM-based models for Arabic sentiment analysis.

7.1 Pretrained Models vs LLM-Based

Results show that the pretrained transformer (CAMEL-MSA), despite having far fewer parameters (110M vs 1.7B–8B), consistently outperforms the evaluated LLMs when the latter are used in a zero-shot, prompt-based way. As shown in Tables 1 and 2, on five of six datasets, CAMEL-MSA achieves substantially higher accuracy and Macro F1-score than the best LLM (e.g., 86.7% vs 44.1% on ArSAS, 47% vs 36.3% on LABR). This indicates that domain-specific fine-tuning provides substantial advantages over general-purpose LLMs for Arabic sentiment analysis.

It is important to note that the evaluated LLMs are used in a zero-shot setting without task-specific fine-tuning. Therefore, the comparison reflects differences in adaptation strategies rather than model size alone. Increasing model size does not consistently improve performance under zero-shot prompting, as observed across several datasets. This suggests that task-specific adaptation plays a more critical role than parameter scale in this setting. Under our protocol, zero-shot prompting alone is often insufficient to match the fine-tuned baseline on several corpora, which has implications for reliable Arabic sentiment classification in resource-sensitive deployments.

On LABR, larger Qwen variants do not monotonically improve Macro F1-score (Table 2); BLOOMZ ordering also varies by dataset.

The ASR-transcribed subset (N = 97) and the synthetic Saudi set (N = 300) are small or synthetically generated; very high scores on some columns warrant caution (limited diversity, possible generator bias). Treat as complementary stress tests, not sole evidence of production readiness.

7.2 Impact of Dataset Characteristics

The performance also shows considerable variation across datasets. Better performance is observed for simpler and more consistent datasets (e.g., Saudi Sentences) compared to complex and dialectal ones (e.g., LABR and ASTD). So, it is essential that the model be tested with relevant and diversified data, which

reflects the complexity of real-life data. The LABR dataset consists of two types of challenges. First, the nature of book reviews. The sentiment in book reviews is often expressed indirectly and is nuanced in the form of critique. As such, the sentiment is not necessarily expressed through direct positive and negative words. As such, the review often contains praise and criticism, metaphor, comparison, and the use of literary words. This sentiment is also implicit and context-dependent, which makes it more difficult for the model to learn than in the case of simpler review datasets. The second challenge is label inconsistencies, as discussed in Section 5.1. This challenge reflects a larger problem in the evaluation of Arabic NLP systems: the absence of standardized preprocessing and label mapping conventions. We recommend that future work explicitly document label mapping strategies and, where possible, use datasets with explicit sentiment labels rather than star ratings.

8. Limitations

This study compares a fine-tuned transformer model with LLMs evaluated in a zero-shot setting. While this reflects common practical usage, it does not isolate the effect of model size from adaptation strategy.

We do not include few-shot prompting or fine-tuning of LLMs, which may improve their performance. Future work should explore these settings for a more comprehensive comparison.

Additionally, some datasets used in this study are small or synthetically generated, which may limit generalizability. Synthetic datasets may introduce biases that do not fully reflect real-world language use.

Finally, LLM performance may be sensitive to prompt design and decoding parameters. Although we standardize these settings, variations may lead to different results.

9. Conclusion

This paper evaluated one pretrained transformer (CAMEL-MSA) and eight large language models on six Arabic sentiment datasets. The results show that the pretrained model substantially outperforms zero-shot LLMs on five of six datasets, with large gaps in accuracy and Macro F1-score (e.g. 47–97% vs 31–44% on the same data), despite having far fewer parameters. We identify key factors affecting performance: dataset complexity, dialectal variation, and label consistency. Another contribution is the identification of inconsistencies in the LABR set’s labeling and their impact on the reproducibility of the evaluation process. Future research directions include: (1) standardizing the evaluation process and the labeling scheme, (2) extending the evaluation to larger models and datasets, (3) examining the impact of

domain-specific pre-training on model performance, and (4) improving the evaluation metrics for multilingual and multi-dialect content.

10. References

- Alahmadi, D. (2025). Human Versus AI: A Comparative Study of Zero-Shot LLMs and Transformer Models Against Human Annotations for Arabic Sentiment Analysis. *International Journal of Advanced Computer Science and Applications*, 16(8). (Match pages/DOI to the published PDF.)
- Alduailej, A., & Alothaim, A. (2022). AraXLNet: Pre-trained language model for sentiment analysis of Arabic. *Journal of Big Data*, 9(1), 72. <https://doi.org/10.1186/s40537-022-00625-z>
- Al-Twairish, N., Al-Khalifa, H., & Al-Salman, A. (2017). ArSAS: An Arabic Saudi sentiment analysis dataset. In *Proceedings of the 24th International Conference on Neural Information Processing (ICONIP)*.
- Aly, M., & Atiya, A. (2013). LABR: A large-scale Arabic book reviews dataset. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 494–498). Association for Computational Linguistics. <https://aclanthology.org/P13-2088>
- Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 9–15). European Language Resources Association. <https://aclanthology.org/2020.osact-1.2>
- Antoun, W., Baly, F., & Hajj, H. (2021). AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 196–207). Association for Computational Linguistics. <https://aclanthology.org/2021.wanlp-1.21>
- Gemma Team. (2025). *Gemma 3 technical report*. arXiv Preprint arXiv:2503.19786. <https://arxiv.org/abs/2503.19786>
- Hasan, M. A., Das, S., Anjum, A., Alam, F., Anjum, A., Sarker, A., & Noori, S. R. H. (2024). Zero- and few-shot prompting with LLMs: A comparative study with fine-tuned models for Bangla sentiment analysis. In *Proceedings of LREC-COLING 2024* (pp. 17808–17818).
- Inoue, G., Alhafni, B., Baimukan, N., Bouamor, H., & Habash, N. (2021). The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 92–104). Association for Computational Linguistics. <https://aclanthology.org/2021.wanlp-1.10>
- Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Le Scao, T., Bari, M. S., Shen, S., Yong, Z.-X., Schoelkopf, H., Tang, X., Radev, D., Aji, A. F., Almubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E., & Raffel, C. (2022). *Crosslingual generalization through multitask finetuning*. arXiv Preprint arXiv:2211.01786. <https://arxiv.org/abs/2211.01786>
- Nabil, M., Aly, M., & Atiya, A. (2015). ASTD: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2515–2519). Association for Computational Linguistics. <https://aclanthology.org/D15-1299>
- Polock, S. I. S. (2025). *Sentiment analysis for low-resource languages using large language models with zero-shot, N-shot prompting, and LoRA fine-tuning* [Master's thesis, SIS Polock]. <https://oulurepo.oulu.fi/handle/10024/56755>
- Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., ... Qiu, Z. (2024). *Qwen2.5 technical report*. arXiv Preprint arXiv:2412.15115. <https://arxiv.org/abs/2412.15115>
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., ... Qiu, Z. (2025). *Qwen3 technical report*. arXiv Preprint arXiv:2505.09388. <https://arxiv.org/abs/2505.09388>
- Zouidine, M., & Khalil, M. (2025). Large language models for Arabic sentiment analysis and machine translation. *Engineering, Technology & Applied Science Research*, 15(2), 20737–20742.

GATE-Reranker: A Strong Arabic Cross-Encoder for Document Reranking

Omer Nacar¹ Mohamed Zaytoon^{2,6} Omar Elshehy^{3,6} Khlood Al Jallad^{4,6}

¹Tuwaiq Academy

²Alexandria University, ³Universität des Saarlandes, ⁴Arab International University, ⁶NAMAA Community
o.najar@tuwaiq.edu.sa

Abstract

Arabic information retrieval increasingly relies on multi-stage pipelines in which a fast first-stage retriever produces candidate passages and a neural reranker refines relevance. While transformer cross-encoders deliver strong effectiveness through joint query–passage encoding, multilingual rerankers achieve competitive performance on Arabic benchmarks. However, systematic analysis of calibration, robustness, and deployment behavior in Arabic-specific settings remains limited. We present **GATE-Reranker**, a compact Arabic cross-encoder initialized from an Arabic semantic embedding backbone and fine-tuned on large-scale mMARCO-style Arabic triplets. The model scores each query–passage pair via full self-attention and a lightweight regression head, enabling plug-and-play second-stage reranking for Arabic search and RAG systems. We evaluate on three Arabic benchmarks covering binary relevance discrimination, controlled multi-negative reranking, and large-scale mMARCO evaluation. While remaining competitive with strong multilingual rerankers in ranking effectiveness, GATE-Reranker demonstrates significantly improved calibration and discriminative behavior. These properties translate into more reliable downstream performance in retrieval and RAG pipelines, while maintaining low GPU memory and latency on a Tesla T4.

Keywords: Arabic, information retrieval, reranking, cross-encoder, mMARCO

1. Introduction

Document reranking has become a central component of modern Information Retrieval (IR) systems. In multi-stage retrieval pipelines, a first-stage retriever efficiently narrows down a large corpus to a manageable candidate set, after which a more expressive reranker refines the final ranking. Cross-encoder architectures, built upon large pre-trained language models, have emerged as a dominant reranking paradigm due to their ability to jointly encode query–document pairs and leverage full self-attention for fine-grained semantic interaction (Déjean et al., 2024). By modeling contextualized token-level relationships across the query and the document, cross-encoders consistently achieve strong ranking effectiveness. Alternative interaction mechanisms, such as late-interaction models (e.g., ColBERT) and listwise LLM-based rerankers, aim to balance effectiveness and efficiency, but often introduce additional computational cost or complexity.

Most high-performing rerankers are trained primarily on English-centric datasets such as MS MARCO (Bajaj et al., 2016) and evaluated on widely adopted benchmarks like BEIR (Thakur et al., 2021). While multilingual models nominally support Arabic, they are rarely optimized specifically for its linguistic properties, including rich morphology, orthographic variation, and syntactic flexibility.

In this work, we introduce **GATE-Reranker**, a

compact Arabic cross-encoder reranker trained on large-scale Arabic MS MARCO-style triplets. Our learning formulation follows *pointwise relevance regression*: each query–passage pair is assigned a continuous score, and the model is optimized to score relevant pairs higher than non-relevant pairs. The model processes the query and passage jointly within a single input sequence, enabling full cross-attention and contextual interaction across segments. Training is conducted using curated Arabic triplets with controlled positive-to-negative sampling, and optimization is performed using mixed-precision to ensure stable and efficient training.

Rather than proposing a new interaction mechanism, our contribution lies in establishing a robust, reproducible, and empirically validated Arabic cross-encoder baseline, together with systematic Arabic-focused evaluation. We evaluate across multiple Arabic reranking benchmarks spanning binary relevance, controlled multi-negative reranking, and large-scale mMARCO evaluation. We further analyze sensitivity to retrieval depth and report efficiency measurements on a Tesla T4 GPU, allowing direct comparison of effectiveness–latency trade-offs. Overall, our results show that an Arabic-specialized cross-encoder can achieve competitive effectiveness relative to strong multilingual rerankers while offering favorable deployment characteristics for Arabic retrieval and RAG pipelines.

By grounding our study in systematic experimentation and Arabic-specific evaluation, we aim to provide a solid empirical foundation for Arabic neural

reranking, showing that Arabic-specialized cross-encoders can achieve competitive effectiveness while offering favourable efficiency trade-offs for deployment.

Approach	Objective	Interaction	Practical Limitation
BERT rerankers (Nogueira and Cho, 2019)	Pointwise	Cross-encoder	High per-pair inference cost
ColBERT (Khattab and Zaharia, 2020)	Pairwise	Late interaction	Multi-vector indexing/storage overhead
Jina reranker v3 (Wang et al., 2025)	Listwise	Joint attention	Cost grows with list/context length
RankGPT (Huang et al., 2025)	Listwise	Generative	Prompt and ordering sensitivity
BGE reranker v2-m3 (Chen et al., 2024)	Pointwise	Cross-encoder	General-purpose; not Arabic-optimized
GATE-Reranker (ours)	Pointwise	Cross-encoder	Second-stage reranking (top- k)

Table 1: Comparison of representative reranking approaches.

2. Related Work

Document reranking methods are typically characterized by their training objectives and query–document interaction mechanisms. Table 1 summarizes representative approaches, highlighting trade-offs between effectiveness and efficiency.

2.1. Learning-to-Rank Paradigms

Reranking approaches are commonly categorized into pointwise, pairwise, and listwise methods. Pointwise models predict absolute relevance scores for each query–document pair, while pairwise approaches learn relative preferences between documents (e.g., RankNet (Burgess et al., 2005), PRP (Qin et al., 2024)). Listwise methods optimize ranking metrics over multiple candidates simultaneously (Wang et al., 2013), often leveraging large language models such as Jina Reranker v3 (Wang et al., 2025) and RankGPT (Huang et al., 2025). While listwise approaches can model richer interactions, they typically incur higher computational cost and increased sensitivity to input ordering.

2.2. Neural Ranking Architectures

Neural rerankers are commonly implemented as bi-encoders or cross-encoders. Bi-encoders independently encode queries and documents into fixed representations, enabling efficient large-scale retrieval but limiting token-level interaction (e.g., DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2021)). In contrast, cross-encoders jointly encode

query–document pairs with full self-attention, enabling fine-grained interaction and improved ranking accuracy (e.g., BERT rerankers (Nogueira and Cho, 2019), MonoBERT (Nogueira et al., 2019)), at the cost of higher inference complexity.

2.3. LLM-Based Rerankers

Recent work explores LLM-based reranking using generative or long-context formulations. Generative approaches such as RankGPT (Huang et al., 2025) produce ordered outputs via prompting, while long-context models such as jina-reranker-v3 (Wang et al., 2025) jointly process multiple candidates. Although effective, these methods are computationally expensive and sensitive to prompt design, while traditional cross-encoders remain competitive and more efficient in production settings (Déjean et al., 2024).

2.4. Multilingual and Arabic Reranking

Recent multilingual rerankers, such as BGE v2-m3 (Chen et al., 2024; Li et al., 2023) and Qwen-based models (Zhang et al., 2025), demonstrate strong performance across languages, including Arabic. However, most prior work focuses primarily on ranking effectiveness, with limited analysis of calibration, robustness, and deployment behavior in Arabic-specific settings. Our work addresses this gap by emphasizing calibration-aware evaluation and reliability-focused analysis.

Arabic representation backbones. The quality of the underlying encoder is a key factor in reranking performance. Nacar et al. (2025) introduce GATE, a General Arabic Text Embedding model trained with multi-task objectives and Matryoshka representation learning. We initialize our reranker from a GATE-based checkpoint and adapt it to cross-encoder reranking via end-to-end fine-tuning on Arabic mMARCO-style triplets.

3. Methodology

This section presents the architecture, learning formulation, data construction strategy, and optimization procedure used to train **GATE-Reranker**. The proposed system follows a transformer cross-encoder design trained with *pointwise relevance regression* on positive/negative query–passage pairs derived from Arabic triplets. An overview of the architecture is illustrated in Figure 1.

3.1. Model Architecture

GATE-Reranker is implemented as a transformer-based cross-encoder. Given a query q and a pas-

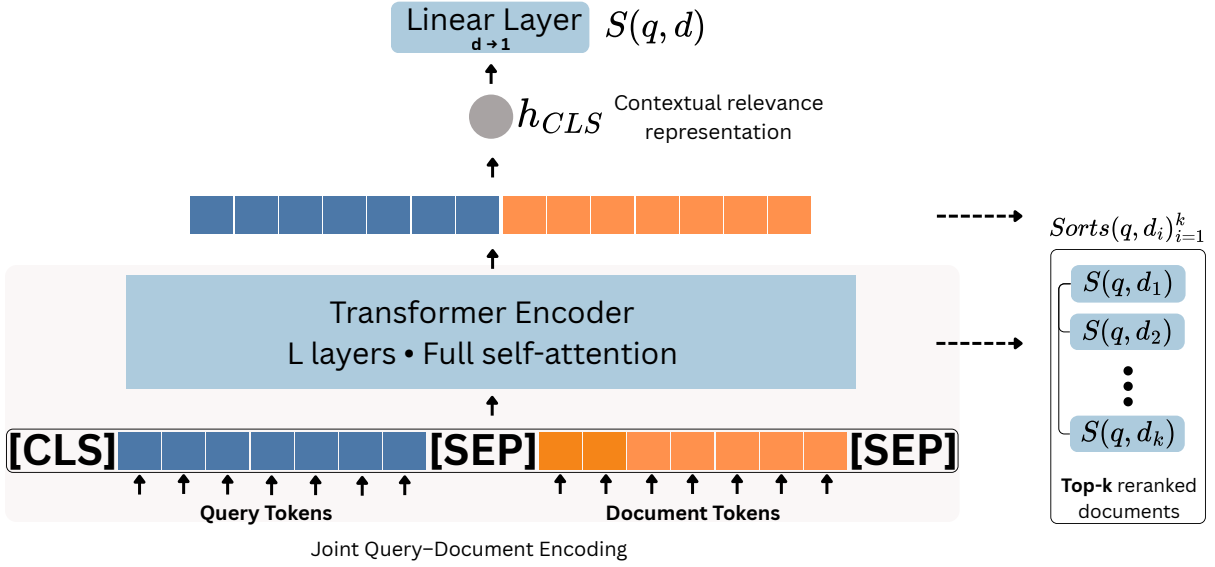


Figure 1: Architecture of GATE-Reranker. The query and document are concatenated and jointly encoded using a transformer-based cross-encoder with full self-attention. The contextualized representation of the [CLS] token is passed through a linear regression head to produce a relevance score $s(q, d)$. Candidate documents are reranked by sorting predicted scores within the top- k set.

sage d , the model constructs a single input sequence:

$$x = [\text{CLS}] \, q \, [\text{SEP}] \, d \, [\text{SEP}]. \quad (1)$$

The concatenated sequence is processed by a transformer encoder composed of L layers with full self-attention. Unlike bi-encoder architectures that encode queries and documents independently, the cross-encoder jointly models token-level interactions across both segments. This enables contextual alignment between query and document tokens through shared attention mechanisms.

Let $H = \text{Encoder}_\theta(x)$ denote the contextualized hidden representations produced by the transformer. The final hidden state corresponding to the [CLS] token, denoted by h_{CLS} , is used as a holistic relevance representation:

$$h_{\text{CLS}} = H_0. \quad (2)$$

A linear regression head then maps this representation to a scalar relevance score:

$$s(q, d) = W h_{\text{CLS}} + b, \quad (3)$$

where W and b are learnable parameters. The resulting score $s(q, d)$ reflects the model’s estimate of semantic relevance between the query and the passage.

Backbone initialization from GATE. We initialize the encoder from a GATE-based Arabic embedding model trained with multi-task objectives (Nacar et al., 2025).

3.2. Training Objective

We adopt a *pointwise* learning formulation expressed as binary relevance regression. For each query, we construct positive and negative query–passage pairs. Positive pairs are labeled with $y = 1$, while negative pairs are labeled with $y = 0$.

Given a mini-batch of N training instances $\{(q_i, d_i, y_i)\}_{i=1}^N$, the model is optimized using the mean squared error (MSE) loss:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (s(q_i, d_i) - y_i)^2. \quad (4)$$

This objective encourages higher scores for relevant passages and lower scores for non-relevant ones. At inference time, ranking is performed by sorting passages according to their predicted scores.

3.3. Justification of Pointwise Regression

We adopt a pointwise regression formulation with MSE loss for its simplicity, stability, and compatibility with score calibration. Unlike pairwise or listwise objectives that optimize relative ordering, pointwise regression directly models absolute relevance scores, enabling better calibration and interpretability.

While pairwise and listwise approaches can improve ranking performance, they often produce uncalibrated scores that are harder to interpret and threshold in practice. In contrast, our objective pri-

critiques score reliability and deployment readiness, rather than purely optimizing ranking metrics.

As shown in Section 6, this choice leads to well-separated and calibrated score distributions, which are critical for downstream applications requiring threshold-based decisions.

3.4. Training Data Construction

Training data is derived from Arabic MS MARCO-style triplets (Bajaj et al., 2016) consisting of a query q , a relevant passage d^+ , and a non-relevant passage d^- . For each triplet, we generate two training instances:

$$(q, d^+, 1), \quad (q, d^-, 0). \quad (5)$$

We employ a balanced positive-to-negative sampling ratio of 1:1, ensuring stable optimization and preventing negative dominance during training. The data is constructed from curated Arabic queries and passages within the mMARCO collection. A separate development subset is maintained for intermediate evaluation during training.

The model is trained with a maximum sequence length of 512 tokens and a batch size of 8. The relatively small batch size accommodates the computational cost of joint encoding at full context length. We train for a single epoch to avoid overfitting and preserve the semantic structure of the pretrained backbone. Empirically, additional epochs did not yield consistent improvements and occasionally degraded calibration performance.

Automatic Mixed Precision (AMP) is enabled to improve computational efficiency and reduce memory footprint without compromising numerical stability. During training, periodic evaluation is conducted using a cross-encoder reranking evaluator constructed from held-out query–positive–negative samples.

Training was conducted using PyTorch and SentenceTransformers. The model was fine-tuned with a batch size of 8, maximum sequence length of 512 tokens, and trained for one epoch using the AdamW optimizer with a learning rate of $2e-5$. Mixed precision (AMP) was enabled for efficiency. All experiments were conducted on a single Tesla T4 GPU.

4. Experiments

This section describes the evaluation setup used to assess GATE-Reranker. We detail the benchmarks, comparison models, evaluation metrics, and experimental protocol, followed by an ablation analysis designed to isolate the effect of retrieval depth on Arabic reranking.

4.1. Benchmarks

We evaluate on three Arabic reranking benchmarks covering both large-scale retrieval and controlled reranking settings.

The first benchmark is the Arabic portion of the mMARCO development set. This dataset follows the MS MARCO reranking protocol, where each query is associated with a candidate set produced by a first-stage retriever over the full Arabic collection. We use this benchmark to assess reranking behavior at scale (55,560 evaluation queries over a collection of approximately 8.8M passages).

The second benchmark, *Arabic-Reranking-Triplet-5-Eval*, consists of 500 Arabic queries, each paired with one relevant passage and four non-relevant passages. This controlled multi-negative setup evaluates whether a reranker can consistently place the positive passage above several competitive distractors under a fixed candidate set size ($k = 5$).

The third benchmark, *Ar-Reranking-Eval*, is a binary relevance dataset of 500 samples in which each query is paired with candidate documents labeled as relevant (1) or non-relevant (0). This benchmark primarily evaluates pair-level relevance discrimination and serves as a sanity check for Arabic query–passage matching.

Together, these datasets provide complementary perspectives: large-scale realistic reranking (mMARCO), controlled multi-negative competition (Triplet-5), and binary discrimination (Ar-Reranking-Eval).

4.2. Compared Models

Our primary contribution in this paper is **GATE-Reranker-V1**, which is trained using the methodology described in Section 3 and is designed to be a compact Arabic cross-encoder suitable for practical second-stage reranking. We additionally report results for a higher-capacity scaling variant of the same training recipe, denoted **GATE-Reranker-LC**. We treat GATE-Reranker-LC as an internal scaling variant rather than a separate competing method, enabling an explicit effectiveness–efficiency comparison within the same Arabic reranker family while keeping the narrative centered on GATE-Reranker-V1.

For external comparison, we evaluate publicly available rerankers spanning multilingual and Arabic-focused cross-encoder models, including mMARCO MiniLM (`cross-encoder/mmarco-mMiniLMv2-L12-H384-v1`), BGE-Reranker-v2-m3 (`BAAI/bge-reranker-v2-m3`), Mizan-Rerank-v1 (`ALJIACHI/Mizan-Rerank-v1`), lightblue/lb-reranker-0.5B-v1.0, and mxbai-rerank-base-v2.

4.3. Evaluation Metrics

We report standard Information Retrieval metrics: Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and nDCG@10. For controlled multi-negative settings, correctness at the top rank is also considered when informative. All metrics are computed consistently across models.

4.4. Evaluation Protocol

All models are evaluated in a pure reranking setting. For each query q and its candidate set $\{d_i\}_{i=1}^k$, the reranker computes relevance scores independently:

$$s_i = s(q, d_i), \quad (6)$$

and produces the final ranking by sorting scores in descending order:

$$\text{Rank}(q) = \text{Sort}(\{s(q, d_i)\}_{i=1}^k). \quad (7)$$

For *Arabic-Reranking-Triplet-5-Eval*, $k = 5$ by construction. For *Ar-Reranking-Eval*, scoring is performed directly on labeled query–document pairs. For mMARCO Arabic dev, candidate sets are produced by a first-stage retriever; in the ablation study below, we use an oracle-controlled construction to ensure the relevant passage is present and to isolate reranker behavior from retriever recall.

4.5. Ablation Study: Effect of Retrieval Depth

On mMARCO Arabic dev, we conduct an ablation study that isolates the effect of varying the number of candidates reranked (k). For each query, we assume a single positive (relevant) passage and treat all remaining candidates as negatives (irrelevant). We evaluate reranking under $k = 2$ and $k = 3$ by pairing the positive passage with one or two negatives sampled from the retrieval pool. This setup enables a controlled analysis of how additional negatives and retrieval depth influence the model’s ability to distinguish relevant from irrelevant passages.

5. Results

This section reports the effectiveness and efficiency of our Arabic reranker family across multiple benchmarks. We focus on **GATE-Reranker-V1** as the primary model described in this paper, and additionally report a higher-capacity scaling variant, denoted **GATE-Reranker-LC**. We treat it as a scaling variant of the same training recipe to analyze the accuracy–efficiency trade-off, while keeping the paper centered on GATE-Reranker-V1. We then present a controlled retrieval-depth ablation on mMARCO Arabic dev, a GPU efficiency study,

Model	MRR	MAP	nDCG@10
GATE-Reranker-V1	1.000	1.000	1.000
GATE-Reranker-LC	1.000	1.000	1.000
mMARCO-mMiniLMv2	1.000	1.000	1.000
BGE-Reranker-v2-m3	1.000	1.000	1.000
Mizan-Rerank-v1	0.998	0.998	0.998
mxbai-rerank-base-v2	0.991	0.990	0.993
lb-reranker-0.5B	0.942	0.943	0.958

Table 2: Results on Ar-Reranking-Eval.

and domain-oriented results from the Arabic RAG Leaderboard (Mohaned A. Rashad, 2025). Importantly, we observe that multilingual rerankers, while competitive in ranking metrics, exhibit severe calibration issues and near-zero discriminative margins (Section 6), limiting their reliability in downstream retrieval pipelines.

5.1. Ar-Reranking-Eval

Table 2 reports results on *Ar-Reranking-Eval*. Several cross-encoder rerankers achieve ceiling performance, including our models (**GATE-Reranker-V1** and **GATE-Reranker-LC**) as well as strong multilingual baselines (mMARCO-mMiniLMv2 and BGE-Reranker-v2-m3). This saturation indicates that the benchmark provides limited ranking ambiguity for modern rerankers—the relevant passage is often highly separable from the non-relevant candidates.

While *Ar-Reranking-Eval* therefore serves as a useful sanity check that confirms the effectiveness of our Arabic reranker family under binary relevance conditions, it is not sufficiently discriminative for comparing top-tier systems. For this reason, we rely on the more competitive multi-negative evaluation (Triplet-5) and the large-scale mMARCO setting to characterize robustness and practical trade-offs.

Model	MRR	MAP	nDCG@10
mMARCO-mMiniLMv2	0.999	0.999	0.9993
BGE-v2-m3	0.999	0.999	0.9993
GATE-Reranker-LC	0.997	0.997	0.9978
GATE-Reranker-V1	0.996	0.996	0.9970
Mizan-Rerank-v1	0.9799	0.9799	0.9850
lb-reranker-0.5B	0.5610	0.5610	0.6700
mxbai-rerank-base-v2	0.4724	0.4724	0.6019

Table 3: Results on Arabic-Reranking-Triplet-5-Eval (500 queries, $k = 5$).

5.2. Arabic-Reranking-Triplet-5-Eval

Table 3 reports results on *Arabic-Reranking-Triplet-5-Eval*, a controlled multi-negative benchmark ($k = 5$) that better separates model quality. Unlike the saturated binary setting, this benchmark reveals meaningful differences among rerankers. Strong multilingual cross-encoders such as mMARCO-mMiniLMv2 and BGE-v2-m3 achieve the highest effectiveness, while our Arabic reranker family

Model	MRR	MAP	nDCG@10
<i>k = 2 (oracle-controlled)</i>			
BGE-v2-m3	0.99933	0.99931	0.99941
GATE-Reranker-LC	0.99877	0.99873	0.99898
mMARCO-mMiniLMv2	0.99842	0.99834	0.99871
GATE-Reranker-V1	0.99825	0.99814	0.99857
Mizan-Rerank-v1	0.95817	0.95717	0.96860
lb-reranker-0.5B	0.76811	0.76564	0.82786
mxbai-rerank-base-v2	0.69286	0.69288	0.77396
<i>k = 3 (oracle-controlled)</i>			
BGE-v2-m3	0.99922	0.99919	0.99932
GATE-Reranker-LC	0.99821	0.99816	0.99856
GATE-Reranker-V1	0.99792	0.99779	0.99831
mMARCO-mMiniLMv2	0.99745	0.99734	0.99797
Mizan-Rerank-v1	0.95203	0.95082	0.96387
lb-reranker-0.5B	0.65198	0.64985	0.74061
mxbai-rerank-base-v2	0.51829	0.51697	0.64103

Table 4: mMARCO Arabic dev ablation showing the effect of retrieval depth ($k = 2$ vs. $k = 3$).

(**GATE-Reranker-V1** and **GATE-Reranker-LC**) follows closely with consistently high performance. In contrast, more general rerankers (e.g., mxbai-rerank-base-v2 and lb-reranker) degrade substantially, suggesting that Arabic relevance modeling under multi-negative competition benefits from Arabic-focused training signals.

Crucially, the gap between **GATE-Reranker-V1** and the best-performing models is small, supporting the central motivation of this work: a dedicated Arabic cross-encoder can achieve near state-of-the-art reranking quality while remaining practical under deployment constraints.

5.3. Ablation: Retrieval Depth on mMARCO Arabic Dev

Table 4 summarizes retrieval-depth ablation on mMARCO Arabic dev. Increasing k from 2 to 3 introduces an additional negative passage per query, increasing ranking difficulty. Top rerankers remain stable, whereas weaker models show larger degradations, indicating sensitivity to additional distractors. Both of our models are robust under this shift, and **GATE-Reranker-V1** remains close to the best multilingual baselines, supporting its suitability for practical multi-stage retrieval where candidate lists often contain multiple hard negatives.

Importantly, comparing **GATE-Reranker-V1** against **GATE-Reranker-LC** isolates the effect of scaling within the same training recipe: the higher-capacity variant provides a modest effectiveness gain, which we later contextualize against its substantially higher inference cost (Section 5.4).

5.4. Inference Efficiency on Tesla T4

Table 5 reports inference efficiency on a Tesla T4 GPU (approximately 14.6 GB VRAM), measured over 250 iterations with 5 warmup steps. The results clarify an important practical trade-off: al-

though scaling can provide small effectiveness gains (as seen for **GATE-Reranker-LC**), the inference cost may become prohibitive for latency-sensitive pipelines.

GATE-Reranker-V1 provides a particularly strong effectiveness–efficiency balance: it achieves near state-of-the-art ranking quality while using substantially less GPU memory and offering low per-pair latency. In contrast, **GATE-Reranker-LC** is significantly slower and more memory-intensive, illustrating that the compact model is better suited for real-time reranking or reranking larger candidate sets. Heavier rerankers such as lb-reranker-0.5B and mxbai-rerank-base-v2 are substantially slower still, making them challenging to deploy at scale.

Model	Latency (ms)	Pairs/s	GPU Mem (MB)
mMARCO-mMiniLMv2	19.62	1274.01	2691.80
GATE-Reranker-V1	28.55	875.80	516.73
Mizan-Rerank-v1	45.95	544.08	3128.02
GATE-Reranker-LC	150.95	165.62	2691.80
BGE-v2-m3	161.59	154.71	3128.02
mxbai-rerank-base-v2	278.08	89.90	3575.04
lb-reranker-0.5B	1115.61	22.41	4091.27

Table 5: Inference efficiency on Tesla T4 (250 iterations; lower latency and memory are better, higher throughput is better).

5.5. Domain-Oriented Evaluation on the Arabic RAG Leaderboard

To assess domain-oriented performance in realistic Arabic RAG scenarios, we additionally report scores from the Arabic RAG Leaderboard (Mohaned A. Rashad, 2025), which evaluates rerankers across multiple domain slices (e.g., global knowledge, tourism, and legal). Table 6 summarizes results for selected rerankers. Notably, **GATE-Reranker-V1** remains competitive across domains despite operating at a shorter context length (512 tokens), whereas several strong multilingual rerankers and our scaling variant **GATE-Reranker-LC** leverage longer context windows (e.g., 8192 tokens). This comparison highlights that Arabic-specialized reranking can deliver strong domain performance even under constrained input budgets, while also offering a clear path for scaling when longer-context deployment is feasible.

6. Error Analysis and Score Calibration

While aggregate ranking metrics provide a high-level comparison, they do not capture how reliably a reranker separates relevant from non-relevant documents. We therefore conduct a detailed error analysis focusing on score calibration, discriminative margin, and false-positive behavior.

Model	Avg	Global	Tourism	Legal
BGE-v2-m3	87.44	81.27	80.96	88.58
GATE-Reranker-LC	85.82	80.18	77.70	87.62
gte-multilingual-reranker-base	85.03	76.76	77.10	85.87
GATE-Reranker-V1	83.96	77.02	79.60	84.41
Mizan-Rerank-v1	76.26	71.68	70.80	77.21

Table 6: Arabic RAG Leaderboard scores for selected rerankers (subset of domains shown for brevity).

6.1. Calibration and Discriminative Power

Table 7 reports calibration and discrimination metrics for **GATE-Reranker** and a strong multilingual baseline (MiniLM-L12-v2).

We observe that **GATE-Reranker** achieves strong calibration (low ECE and Brier score) and a large discriminative margin, while MiniLM exhibits severe miscalibration and near-zero margin. Notably, MiniLM assigns almost identical scores to both positive and negative samples, indicating a collapse of the scoring function.

6.2. Large-Margin Separation

We measure the average margin between positive and negative samples:

$$\Delta = E[s(q, d^+) - s(q, d^-)]. \quad (8)$$

GATE-Reranker achieves a margin of 0.9382, while MiniLM produces a near-zero margin of 0.0002. This difference spans several orders of magnitude, indicating that GATE-Reranker forms a clear decision boundary, whereas MiniLM fails to meaningfully separate relevant from non-relevant documents.

6.3. False Positive Behavior

We further analyze the false positive rate at high recall (FPR@95TPR). GATE-Reranker achieves 0.0, indicating perfect rejection of negative samples under high recall conditions. In contrast, MiniLM reaches 0.6453, meaning that a large proportion of negative documents are incorrectly classified as relevant.

This result highlights a critical limitation of multilingual rerankers: while they may achieve reasonable ranking metrics, they can produce unreliable decisions when used for filtering or threshold-based retrieval.

6.4. Score Distribution Analysis

Distribution Analysis. Figure 2 provides a visual explanation of the observed quantitative differences. GATE-Reranker produces a well-separated

bimodal distribution, with relevant documents concentrated near 1 and non-relevant documents near 0. This separation reflects strong discriminative capacity and enables reliable threshold-based decision-making.

In contrast, MiniLM exhibits a collapsed distribution where both positive and negative samples are concentrated in a narrow region near 1 (right panel, zoomed). This collapse indicates that the model fails to meaningfully separate relevant from non-relevant documents, explaining its near-zero margin and high false positive rate.

Notably, MiniLM assigns nearly identical score distributions to both classes, effectively collapsing the scoring function into a non-discriminative regime.

6.5. Implications for Retrieval Systems

These findings highlight an important distinction between ranking performance and decision reliability. While multilingual rerankers may remain competitive on ranking metrics, they often exhibit poor calibration and limited robustness to topical noise. In contrast, GATE-Reranker produces well-separated and calibrated scores, enabling more reliable relevance estimation. We attribute these gains to the pointwise regression formulation and the Arabic-specific semantic initialization, which together encourage consistent score scaling and stronger separation between relevant and non-relevant documents. As shown in Section 6, this results in significantly larger margins and lower false positive rates compared to multilingual baselines.

These improvements have direct implications for downstream systems such as RAG pipelines. Well-calibrated scores enable the use of global thresholds to filter irrelevant documents before generation, reducing noise propagation and improving answer quality. In contrast, poorly calibrated rerankers may pass irrelevant documents despite correct ranking order, leading to degraded generation and increased hallucination risk. Furthermore, strong score separation allows more effective top- k pruning, where fewer but higher-quality documents are selected for downstream processing. This reduces computational cost while maintaining relevance, improving both efficiency and reliability in end-to-end retrieval systems.

6.6. Effectiveness vs. Reliability Trade-off

Although GATE-Reranker does not always outperform strong multilingual rerankers in ranking metrics, it provides significantly better calibration and discriminative margins. This suggests a trade-off between ranking optimization and score reliability.

Multilingual models may achieve strong ranking performance by leveraging large-scale cross-

Model	ECE ↓	Brier ↓	Margin ↑	FPR@95 ↓
GATE-Reranker	0.0316	0.0077	0.9382	0.0000
MiniLM-L12-v2	0.4995	0.4994	0.0002	0.6453

Table 7: Calibration and discriminative analysis. Lower is better for ECE, Brier score, and FPR@95TPR; higher is better for margin.

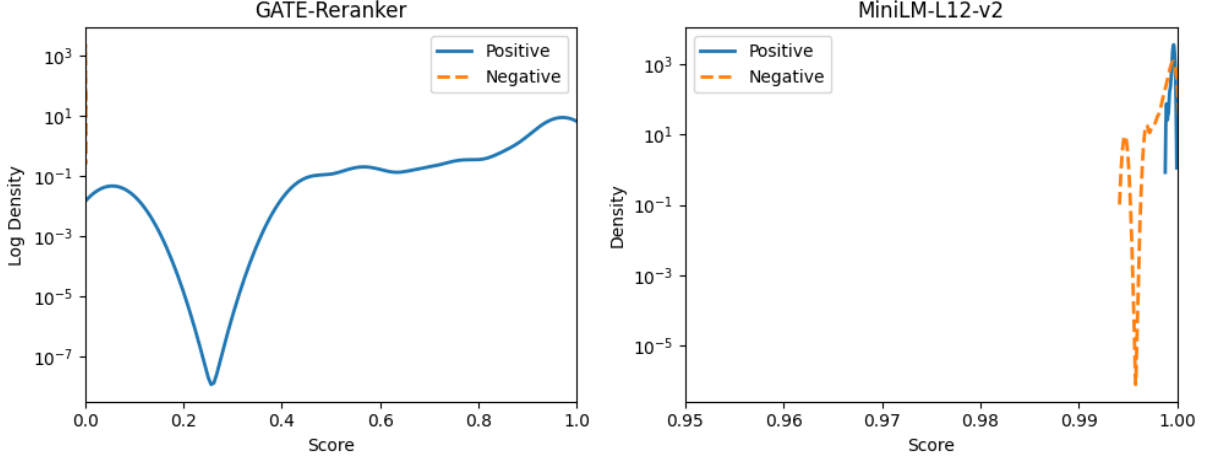


Figure 2: Log-scale score distributions for GATE-Reranker (left) and MiniLM-L12-v2 (right). GATE exhibits a clear bimodal distribution with strong separation between relevant and non-relevant documents. In contrast, MiniLM collapses both classes into a narrow high-score region (zoomed), indicating severe miscalibration and near-zero discriminative margin.

lingual data, but often lack precise score separation, as shown in Section 6. In contrast, GATE-Reranker prioritizes reliable scoring behavior, producing well-calibrated outputs that are more suitable for real-world deployment scenarios.

7. Conclusion

We introduced **GATE-Reranker**, a compact Arabic cross-encoder trained with pointwise relevance regression on Arabic mMARCO-style triplets. Our results across binary, multi-negative, and large-scale Arabic benchmarks show that an Arabic-specialized reranker can achieve competitive effectiveness relative to strong multilingual baselines while maintaining substantially lower inference cost. In particular, GATE-Reranker-V1 provides a favorable effectiveness–efficiency trade-off, combining near state-of-the-art ranking quality with low GPU memory usage and latency. We additionally analyzed a higher-capacity scaling variant to quantify the accuracy–efficiency frontier within the same training paradigm. Our findings suggest that calibration-aware reranking is as important as ranking effectiveness for real-world deployment. Overall, this work establishes a strong and reproducible Arabic reranking baseline and highlights the importance of Arabic-focused evaluation and deployment-aware modeling in neural retrieval systems.

8. Limitations

GATE-Reranker provides strong discriminative power and well-calibrated scores, enabling reliable threshold-based filtering and robust behavior in downstream systems such as RAG, while maintaining a favorable effectiveness–efficiency trade-off. However, the study assumes a two-stage retrieval pipeline and does not evaluate end-to-end recall. The model is also limited by a 512-token context window, which may affect performance on long passages, and some evaluation benchmarks exhibit near-ceiling results, reducing their discriminative value. Additionally, the approach may inherit biases from mMARCO-style training data, including domain imbalance and limited dialectal coverage, as well as from the underlying Arabic embedding backbone, potentially affecting generalization across domains and linguistic variations.

9. References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

- Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. [Learning to rank using gradient descent](#). In *Proceedings of the 22nd International Conference on Machine Learning, ICML '05*, page 89–96, New York, NY, USA. Association for Computing Machinery.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#).
- Hervé Déjean, Stéphane Clinchant, and Thibault Formal. 2024. A thorough comparison of cross-encoders and llms for reranking splade. *arXiv preprint arXiv:2403.10407*.
- Yanhua Huang, Yuqi Chen, Xiong Cao, Rui Yang, Mingliang Qi, Yinghao Zhu, Qingchang Han, Yaowei Liu, Zhaoyu Liu, Xuefeng Yao, Yuting Jia, Leilei Ma, Yinqi Zhang, Taoyu Zhu, Liujie Zhang, Lei Chen, Weihang Chen, Min Zhu, Ruiwen Xu, and Lei Zhang. 2025. [Towards large-scale generative ranking](#).
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.
- Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. [Making large language models a better foundation for dense retrieval](#).
- Hamza Shahid Mohaned A. Rashad. 2025. The arabic rag leaderboard. url<https://huggingface.co/spaces/Navid-AI/The-Arabic-Rag-Leaderboard>.
- Omer Nacar, Anis Koubaa, Serry Sibae, Yasser Al-Habashi, Adel Ammar, and Wadii Boulila. 2025. Gate: General arabic text embedding for enhanced semantic textual similarity with matryoshka representation learning and hybrid loss training. *arXiv preprint arXiv:2505.24581*.
- Rodrigo Nogueira and Kyunghyun Cho. 2019. [Passage re-ranking with bert](#). *ArXiv*, abs/1901.04085.
- Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. [Multi-stage document ranking with bert](#).
- Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. [Large language models are effective text rankers with pairwise ranking prompting](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.
- Feng Wang, Yuqing Li, and Han Xiao. 2025. [jina-reranker-v3: Last but not late interaction for list-wise document reranking](#).
- Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pages 25–54. PMLR.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. [Approximate nearest neighbor negative contrastive learning for dense text retrieval](#). In *International Conference on Learning Representations*.
- Yanzhao Zhang, M Li, D Long, X Zhang, H Lin, B Yang, P Xie, A Yang, D Liu, J Lin, et al. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025.

How Foundation Models behave for Arabic Image Captioning?

Khaoula Dahimi¹, Hadda Cherroun¹, Amel Belabbaci¹ and Abdelhamid Haouhat¹

¹ Laboratoire d'Informatique et de Mathématiques LIM

Amar Telidji University, Laghouat, Algeria

{k.dahimi, hadda_cherroun, a.belabbaci, a.haouhat}@lagh-univ.dz

Abstract

Image captioning plays a crucial role in numerous applications, including educational systems. However, ensuring caption quality remains a significant challenge, particularly for morphologically rich, low-resource languages such as Arabic. We investigate an evaluation of Arabic image captioning using state-of-the-art multimodal foundation models. We systematically assess the performance of leading models—Gemini, Gemma, LLaMA, and Fanar. Our evaluation framework employs a diverse set of metrics spanning rule-based, learnable, visually-grounded, and LLM-based approaches to capture semantic accuracy, linguistic fluency, and hallucination detection. Experiments are conducted on two benchmark datasets: Flickr8k-Arabic and JEEM. Our findings reveal significant performance variations across models and evaluation metrics, highlighting the need for Arabic-specific optimization in multimodal architectures.

Keywords: Multi-Modal Foundation Models, Arabic Image Captioning, Image Caption Evaluation, Rule-based Metrics, Learnable Metrics, LLM-based Metrics, Hallucination Detection.

1. Introduction

The emergence of large-scale foundation models has fundamentally transformed the landscape of natural language processing, with multimodal architectures demonstrating unprecedented performance across diverse tasks (Qin et al., 2024). Vision-language models, including Gemini (Team et al., 2023), Gemma (Team et al., 2025b), and LLaMA (Dubey et al., 2024), have exhibited substantial capabilities in image-to-text generation tasks, particularly in automatic caption generation. Despite these advances, empirical evaluations have predominantly concentrated on high-resource languages, particularly English (Zhu et al., 2023), resulting in limited understanding of model behavior and performance characteristics when applied to morphologically rich, low-resource languages such as Arabic (Haouhat et al., 2025).

The automatic generation of textual descriptions from visual content constitutes a critical component in numerous domains, including assistive technologies for visually impaired users, multimedia content retrieval, and computer-assisted language learning systems (Dongare et al., 2024; Hasnine et al., 2019; Delassi et al., 2025). For Arabic—a language with over 400 million native speakers and significant morphological complexity—the development of robust image captioning systems presents considerable challenges attributable to dataset scarcity (Team et al., 2025a), dialectal variation, and intricate grammatical structures. Within AI-assisted educational frameworks, automatically generated captions offer potential as reference standards for learner out-

put evaluation and language proficiency assessment (Sarto et al., 2025), yet the reliability of such baselines remains underexplored.

This study systematically investigates the performance of a set of state-of-the-art multimodal foundation models—Gemini, Gemma, LLaMA, and Fanar in generating Arabic image captions using the Arabic Flickr8k dataset (ElJundi et al., 2020).

Our investigation employs established quantitative metrics (Papineni et al., 2001; Banerjee and Lavie, 2005; Hessel et al., 2022) to evaluate caption quality, semantic accuracy, and the prevalence of hallucination phenomena (Petryk et al., 2024; Rohrbach et al., 2019).

The investigation pursues two primary objectives: first, to assess whether model-generated captions exhibit sufficient quality and consistency to serve as reliable reference baselines for automated evaluation in AI-driven educational contexts; second, to examine whether existing quantitative metrics adequately capture the quality of generated descriptions, particularly in educational settings involving visual assignments.

The rest of this paper is organized as follows. Section 2 is dedicated to introducing some basic concepts on image captioning evaluation metrics. Section 3 reviews related work on Arabic image captioning. Section 4 presents our experimental methodology, including the selected metrics. Section 5 details the experimental setup, presents the results, and provides a comparative analysis of the targeted models. Finally, we conclude the paper with a summary of our findings and directions for future work.

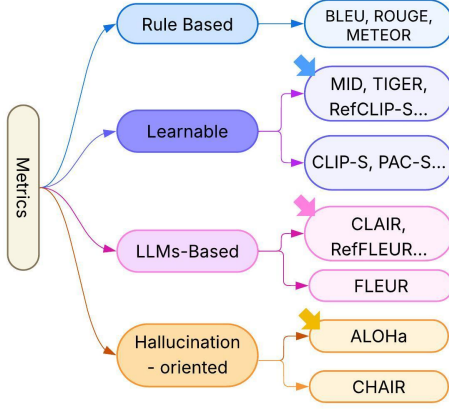


Figure 1: Taxonomy of Image Captioning Evaluation Metrics. Highlighted boxes indicate reference-based metrics that require ground-truth captions for comparison

2. Background

This section surveys the landscape of image captioning evaluation metrics and some existing Arabic image captioning datasets.

2.1. Image Captioning Evaluation Metrics

Image captioning evaluation metrics play a crucial role in assessing the performance of generative models. These metrics measure the quality of the generated text, ensuring that captions are fluent, accurate, and faithful to the image content while penalizing hallucinations or misleading information. Over time, many evaluation metrics have been proposed and refined alongside the evolution of large language models and the image captioning task itself. As illustrated in Figure 1, these metrics are organized into four main categories according to their computational foundations, following the taxonomy proposed by Sarto et al.’s comprehensive survey (Sarto et al., 2025).

Rule Based Metrics

These metrics were originally designed for evaluating machine translation systems. They rely on the principle that “the more the matches, the better the candidate translation.” BLEU (Papineni et al., 2001) measures how many n-grams (words or sequences of words) in the generated text also appear in the reference text, using a weighted average approach. While BLEU focuses primarily on n-gram precision, METEOR (Banerjee and Lavie, 2005) extends the evaluation by also considering recall, stemming, synonyms, and word order, making it more linguistically informed. It operates at the word level, aligning the generated and reference

texts to find the best possible matches, even when the words are not identical (e.g., *run* ↔ *running* or *big* ↔ *large*)

Learnable Metrics

Rule-based metrics depend on surface-level n-gram overlap and may fail when different words conveys the same meaning. Learnable metrics address this limitation by evaluating semantic similarity.

BERTScore (Zhang et al., 2020) metric evaluates the semantic similarity between the generated and reference text (Hanna and Bojar, 2021). They perform this by representing each token with contextual embeddings from a pretrained BERT model and comparing them using cosine similarity. Given reference tokens r and predicted tokens p , using the recall, R_{BERT} , and precision P_{BERT} , BERTScore computes:

$$\text{BERTscore } F_{\text{BERT}} = 2 \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \quad (1)$$

Where :

$$R_{\text{BERT}} = \frac{1}{|r|} \sum_{i \in r} \max_{j \in p} \vec{r}_i^T \vec{j}, \quad P_{\text{BERT}} = \frac{1}{|p|} \sum_{j \in p} \max_{i \in r} \vec{r}_i^T \vec{j}$$

However, these metrics still ignore the visual content and are sensitive to the style and quality of reference captions (Sarto et al., 2025). To address this gap, TIGEr (Jiang et al., 2019) and CLIPScore (Hessel et al., 2022), visually grounded metrics, were introduced. They assess how visually and semantically aligned a caption is with an image. Specifically, for a given image-caption pair. TIGEr (Jiang et al., 2019) computes the matching between textual tokens and image regions using a Text-to-Image Grounding approach. Similarly, CLIPScore (Hessel et al., 2022) uses the CLIP model to generate visual-text embeddings for each image-caption pair and evaluates their alignment via cosine similarity as follows:

$$\text{CLIP\_S}(c, v) = w \cdot \max(\cos(c, v), 0) \quad (2)$$

where: c is the textual CLIP embedding, v the visual CLIP embedding and $w = 2.5$ an empirical value (Hessel et al., 2022).

LLMs Based Metrics

With the emergence of powerful LLMs, metrics such as CLAIR (Chan et al., 2023) and FLEUR (Lee et al., 2024) incorporate them directly into the evaluation pipeline. These metrics convert caption evaluation into a text completion task solved using a structured prompt.

CLAIR (Chan et al., 2023) evaluates how well a candidate caption describes an image by comparing it to reference captions. It outputs both a numerical score and a brief explanation.

FLEUR (Lee et al., 2024) extends this by incorporating the actual image into the evaluation prompt, improving accuracy. It also introduces score smoothing, which uses the token probabilities of each digit generated by the model to produce less noisy and more interpretable scores.

Hallucination Oriented Metrics

Unlike the aforementioned metrics that focus on evaluating caption quality, CHAIR (Rohrbach et al., 2019) and ALOHa (Petryk et al., 2024) are designed specifically to measure object hallucination in generated captions. Hallucination occurs when a caption includes information or objects not actually present in the image (Sarto et al., 2025).

CHAIR (Rohrbach et al., 2019) detects hallucinations by comparing objects mentioned in the generated caption with those found in the reference captions and segmentation labels of the image, allowing it to determine accurately whether an object is truly present. However, CHAIR is limited to the fixed object vocabulary of the MS COCO dataset.

ALOHa (Petryk et al., 2024) addresses this limitation by moving to an open-vocabulary setting. By using LLMs to extract object mentions and semantic similarity to match them. ALOHa provides a more flexible and generalizable way to detect hallucinations.

2.2. Image Captioning Datasets

Arabic image captioning datasets can be categorized into two primary groups:

Translation-Based Datasets comprises datasets derived by translating established English-language benchmarks into Arabic. These include Arabic Flickr8k (ElJundi et al., 2020) and Multilingual COCO (JP, 2025), which adapt their respective originals with Modern Standard Arabic (MSA) captions, and the Arabic Image Captioning (Hennara et al., 2025) dataset, which incorporates translated samples from UCSC-VLAA¹ and Recap-DataComp-1B (Li et al., 2024).

Natively Collected Datasets consists of resources specifically curated by native Arabic speakers to ensure linguistic authenticity and cultural relevance. Notable examples are JEEM (Kadaoui et al., 2025), FutureBeeAI’s Arabic Image Caption (FutureBee AI, 2025) Dataset,

and the ImageEval 2025 dataset from the shared task (Bashiti et al., 2025).

3. Related Work

Many studies have addressed the challenges of Arabic Image Captioning (AIC) using Deep Learning techniques, typically employing CNN-based encoders and LSTM-based decoders. The authors in (ElJundi et al., 2020) use an LSTM sequence model that outperforms traditional English-caption translation approaches. Meanwhile, (Afyouni et al., 2021) introduce a novel architecture that combines object detection with attention-based caption generation.

On the other hand, (Hejazi and Shaalan, 2021) present a comparative study analyzing factors that influence caption generation—such as pre-processing choices, Deep Learning techniques, and feature extraction methods—and offer practical recommendations. Additionally, (Al-Malki and Al-Aama, 2023) suggest that classifying image attributes before feeding them into the decoder can further improve Arabic caption quality.

Only a few studies incorporate transformer-based models. Specifically, (Emami et al., 2022) employ a pre-trained transformer to model relationships between object tags extracted by a CNN encoder. Similarly, (Alsayed et al., 2023) explore the impact of different Arabic text-preprocessing methods alongside various image recognition models using a transformer-based approach.

Authors in (Elchafei and Fashwan, 2025) introduce an interpretable Arabic image captioning approach that retrieves visual concepts using multilingual CLIP models and feeds them as prompts to vision-language decoders. This two-stage design improves semantic grounding and leads to better caption quality than traditional end-to-end models.

Recent efforts have aimed to advance Arabic image captioning through shared tasks and community-driven initiatives. ImageEval’2025 (Bashiti et al., 2025) introduced the first dedicated shared task for Arabic image captioning, combining collaborative dataset creation with competitive model evaluation to address challenges such as cultural specificity, morphological complexity, and dataset scarcity.

4. Methodology

To systematically examine how multimodal foundation models handle the Arabic language in the context of image caption generation, we have drawn out a comprehensive evaluation pipeline fed by images from Arabic datasets, designed to assess model performance under different experimental

¹UCSC-VLAA Lab.

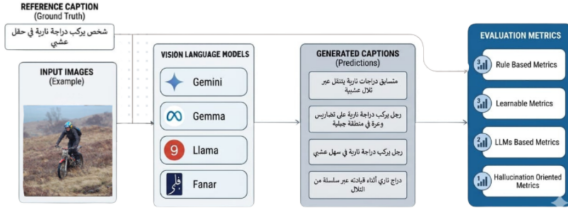


Figure 2: Overview of the comprehensive evaluation pipeline for Arabic image captioning.

conditions and using diverse evaluation criteria and metrics.

The proposed pipeline is illustrated in Figure 2. It comprises four sequential stages: dataset selection and preparation, model-based caption generation, preprocessing, and multi-metric evaluation.

Dataset Selection

A curated set of image-caption pairs from different Arabic image datasets serves as the evaluation corpus, providing sufficient visual and linguistic diversity while maintaining computational feasibility.

Each image is independently processed through multiple models to generate corresponding Arabic captions, which are subsequently compared against human-generated reference captions. To generate captions of similar length, we provide the model with an explicit length prompt. By controlling the length, we can fairly evaluate the models’ ability to produce descriptive captions across datasets with varying caption styles and levels of detail.

Model Selection

In this investigation a panoply of models is targeted, intentionally chosen to represent different architectural paradigms and training methodologies. The selected models are categorized into two primary groups based on their design focus and characteristics of training data. The first category comprises *state-of-the-art (SOTA) multilingual multimodal models*, including *Gemini*, *Gemma*, and *LLaMA*, which have demonstrated exceptional performance across multiple languages and vision-language tasks. These models are predominantly trained on large-scale multilingual corpora that include Arabic among other languages, but without specific optimization for Arabic linguistic features. The second category encompasses *Arabic-specific multimodal models*, which have been either pre-trained or fine-tuned specifically on Arabic datasets to capture the morphological richness, dialectal variations, and syntactic structures inherent to the Arabic language.

This categorization enables comparative analysis between general-purpose multilingual models and language-specific architectures, providing insights into whether specialized Arabic training yields superior performance for Arabic caption generation.

Caption Preprocessing

Before evaluation, all model-generated captions undergo a systematic preprocessing phase to ensure consistency and comparability across models. This preprocessing step is critical to ensure that the subsequent evaluation focuses purely on the semantic and linguistic quality of the generated content rather than superficial formatting differences. The caption are normalized. This process involved removing unnecessary elements and eliminating any non-Arabic words to ensure consistency in the analysis.

Multi-Metric Evaluation Framework

The resulted image captions are then evaluated using a comprehensive set of metrics designed to capture different dimensions of caption quality. These metrics are multi-purpose, each addressing specific aspects of the caption generation task. We employ both *reference-based metrics* that compare generated captions against human-written references and *reference-free metrics* that assess caption quality based on intrinsic properties.

This multi-purpose evaluation strategy provides a holistic assessment of model performance, enabling identification of specific strengths and weaknesses in Arabic caption generation across different model categories and experimental conditions.

5. Experiments and Results

To achieve our research objectives, we designed a comprehensive series of experiments with two main goals in mind. The first goal is to identify which foundation models produce the most accurate and high quality Arabic image captions based on our evaluation metrics. We systematically compare outputs from SOTA multilingual models and Arabic-specific architectures, testing them under different conditions—both with and without prompting. The second goal addresses a critical question: which metrics are best suited for evaluating Arabic linguistic quality in the context of language learning? Our experimental design carefully examines two types of metrics: those that rely on human-written reference captions for comparison, and those that can assess caption quality independently. This distinction matters greatly in real-world educational settings, where reference captions may not always be available.

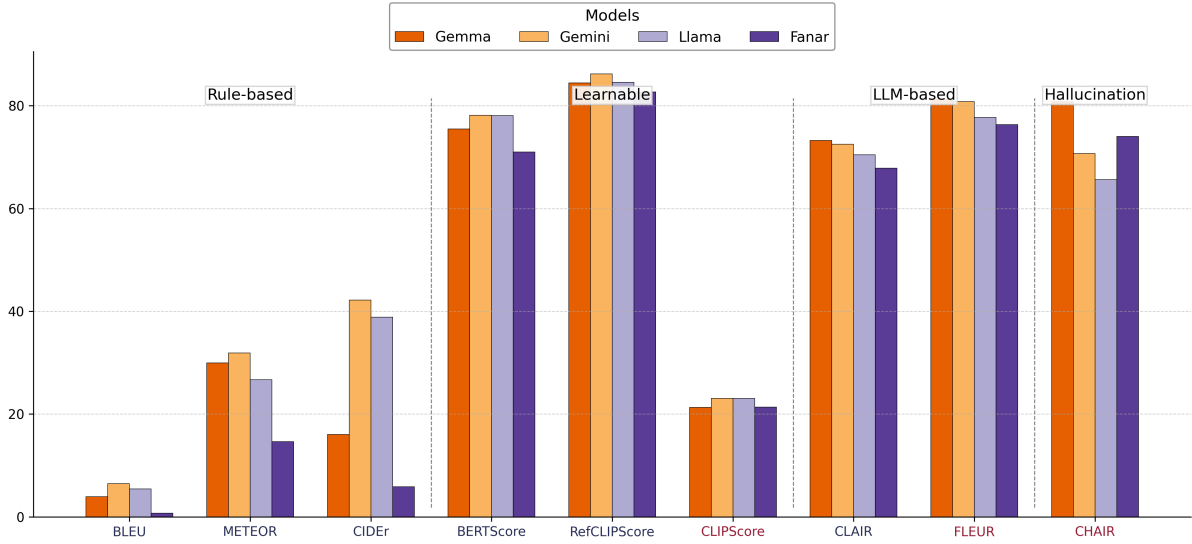


Figure 3: Performance Assessment of Multimodal Models across Nine Metrics on Arabic Flickr8K dataset

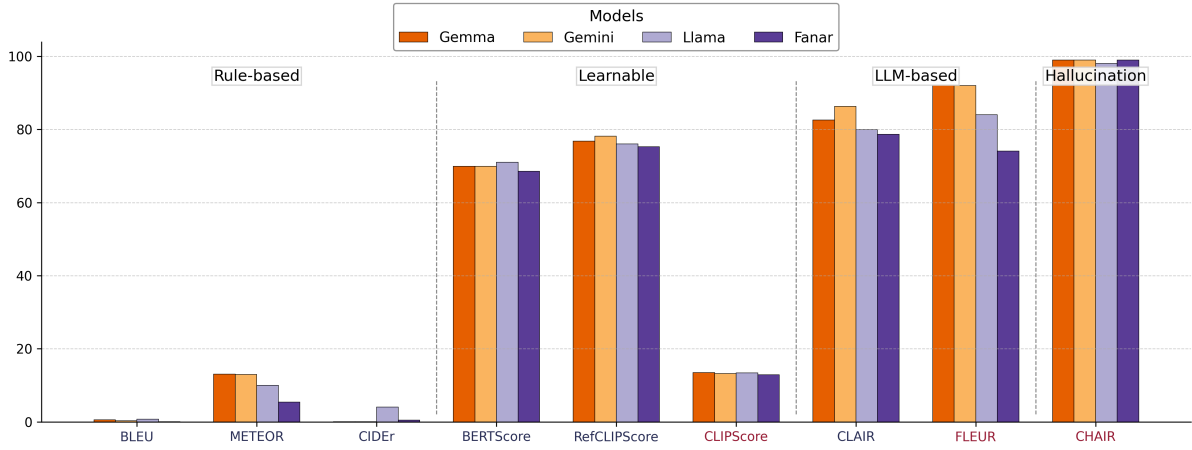


Figure 4: Performance Assessment of Multimodal Models across Nine Metrics on JEEM dataset

We start by describing the experiment testbed namely the used dataset, foundation models and metrics. All our experimentation scripts are publicly accessible on [GitHub](#).

5.1. Datasets

We utilized two distinct Arabic image captioning datasets: the Arabic version of the well-known Flickr8k dataset, and the natively collected JEEM dataset, which covers a broader range of domains and contains longer, more detailed descriptions that convey richer information about the images. Both resources provide standardized collections of images paired with descriptive captions, enabling consistent evaluation of vision-language models. A detailed summary of these datasets is presented in Table 1. It shows that JEEM has a large domain coverage. Its captions are more long, so that they give more detailed on the image.

5.2. Targeted Vision Language Models

Concerning the SOTA models, we have used:

Gemini-2.5-Flash²: Developed by Google AI, it is a generative model designed for high-quality text generation.

Gemma-3-27B-IT³: It is an advanced generative model introduced by Google AI (Team et al., 2025b). It is widely recognized for its instruction-tuned capabilities in text generation.

LLaMA-4-Scout-17B-16E-Instruct⁴: Developed by Meta AI, it is a state-of-the-art generative model commonly used for various language generation tasks.

Concerning the Arabic multimodal models and due to their scarcity, we have narrowed the study

²API version of the model [Gemini Model](#)

³[Gemma Model](#): the API version of the model

⁴[LLaMa Model](#): the API version of the model

Table 1: A Detailed Overview of Arabic Vision-Language Datasets Utilized in This Study

Dataset	Size	Captions per Image	Avg. Words	Resolution	Domain	Experimental Subset
Arabic Flickr8k	8,092	3 references	9–12	640×480	Everyday objects, people, animals, scenes	1,173
JEEM	2,196	1 reference	44–63	Variable	Places, events, arts, nature, education, transport, food, trade, technology, characters, games	1,704

to Fanar Vision version **Fanar-Oryx-IVU-1**⁵. It is developed by Fanar.ai, and represents an Arabic multimodal model designed for visual and linguistic understanding, including image captioning.

5.3. Results and Discussion

The performances of model-generated Arabic image captions are reported across all evaluation metrics in Figures 3 and 4.

The evaluation reveals clear differences among multimodal LLMs across lexical, semantic, and hallucination-sensitive metrics.

The overall performance of the models is quite similar, with Gemini & Gemma achieving the highest scores across both datasets. Additionally, Fanar model performs poorly compared to the others, which is unexpected for an Arabic model, this may be due to the diversity of the selected dataset, and the Fanar model may not have been trained on it.

Regarding the **rule-based metrics**, the models show low scores for the Arabic Flickr8k dataset, suggesting a limited lexical overlap between the generated captions and the reference texts. There are also challenges in capturing semantic adequacy and linguistic variation, especially for morphologically rich languages such as Arabic. The scores are even worse for the JEEM dataset, as expected, due to its longer descriptions.

For the **learnable metrics**, BERTScore and RefCLIPScore evaluate the similarity between generated captions and reference texts. The models achieve high scores on both datasets, indicating strong alignment with the references. This suggests that the generated captions effectively paraphrase or semantically match the human-provided captions.

In contrast, CLIPScore measures the similarity between the image and the generated caption within CLIP’s joint embedding space. The models obtain relatively low CLIPScores on both datasets. This may reflect limitations in CLIP’s representations, such as sensitivity to non-internet-style phrasing, domain mismatch, or reduced robustness to long and complex sentences (Zur et al., 2024; Chen et al., 2025). In summary, although

the generated captions appear strong according to BERTScore and RefCLIPScore, a low CLIPScore does not necessarily indicate poor accessibility quality.

For the **LLM-based metrics**, the models achieve high scores on CLAIR and FLEUR, demonstrating their ability to generate descriptive, visually grounded captions that closely reflect the image content. Notably, Gemma and Gemini consistently outperform the other models across both datasets.

Metric Reliability Analysis

We have observed that all models exhibit relatively poor performance on the CHAIR metric (where lower values indicate better performance). A deep manual analysis of over 100 generated descriptions revealed consistently captions’ good quality, suggesting a discrepancy between the metric’s hallucination indicators and the actual output quality, as illustrated in Table 2.

The misleading CHAIR scores may explain the seemingly poor performance of the models, as the metric is built on a limited object vocabulary derived from the MS COCO dataset, as illustrated in the first image of Table 2. In addition, CHAIR relies on strict object matching and does not account for deeper caption understanding. Consequently, it may classify contextually inferred objects—rather than physically visible ones—as hallucinations, as shown in the second image in Table 2.

Poor CHAIR performance should therefore be attributed to metric limitations rather than model failure.

Linguistic and Semantic Error Analysis of Generated Captions

Table 3 illustrates representative examples of linguistic errors observed in model-generated Arabic captions. Two primary error types are identified. The first involves object-level mismatches combined with grammatically incorrect word usage, as seen in the first example where the generated caption incorrectly refers to horses and uses a morphologically ill-formed term. The second error type concerns grammatical agreement viola-

⁵Fanar Model : the API version of the model

Image	Generated Caption	CHAIR Score	Observation
	رجل يتسلق الثلج بواسطة معدات التسلق، مرتدياً قفازات و أحذية الثلوج و قبعه حمراء En: A man climbs the snow using climbing equipment, wearing gloves , snow boots , and a red hat .	1	These objects are visually present in the image.
	طفله صغيره تتسلق شبكة جبال حمراء في ملعب وتنتظر الي الكاميرا En: A little girl climbs a red rope net in a playground and looks at the camera .	1	The object is not visually detected, although it is implicitly inferred from the photographic context.

Table 2: Illustration of false hallucination cases for generated captions.

Image	Reference	Incorrect caption	CLAIR Score	Observation
	ثلاثة كلاب من أحجام مختلفة في فناء خارجي En: Three dogs of different sizes in an outdoor courtyard.	خيول صغيرة نشطة و ممرحلة في مساحة خضراء En: Small, active and horses in a green open space.	0.8	The caption contains an object-level mismatch and also includes a grammatically incorrect word.
	كلبان بنيان يركضان جنباً إلى جنب En: Two brown dogs are running side by side.	كلبان يركضون في الحديقة En: Two dogs are running in the park.	0.8	The caption contains a grammatical agreement error related to the Arabic dual form, a morphological category absent from English.
	صبيان يركضان و يلعبان بالماء En: Two boys are running and playing with water.	طفل يستمتع باللعب بالماء في حديقة En: A child is enjoying playing with water in a garden.	0.9	The caption incorrectly represents the number of subjects.
	طبق من الكثافة التابلسية الناعمة التي تقدم علي شكل مزروعات بحسب الوزن، وهي من أشهر الحلويات في الأردن، وتتميز بوجهها المحمر المغطى بطبقة من الفستق الحلبي المقروم للزينة، يظهر بعض القطر الذي يضافي عليها نكهة حلوة غنية إضافة إلى طبقة الجينة الدائبة الموجودة تحت العجينة.	طبق من الحلوى الشرقية، تحديداً البقلاوة، مقدمة على طبق أبيض دائري، يقع الطبق على سطح طاولة ذات ملمس مميز، ويبدو أن الإضاءة في المكان خافتة ودافئة. البقلاوة قطعة مربعة الشكل، ذات لون ذهبي بني، تبدو مقرمشة وملينة بالطبقات الرقيقة من العجين، مزينة بكمية صغيرة من الفستق الحلبي المقروم.	0.8	The caption contains an error in object identification, as it incorrectly labels the dessert.

Table 3: Examples of grammatical and semantic errors in automatically generated captions that go unpenalized by evaluation metrics.

tions, where the generated caption fails to produce the correct verb form with respect to number, as demonstrated in the second example. Despite these linguistic inaccuracies, both captions receive a relatively high CLAIR score of 0.8, indicating that the metric assigns favorable scores without penalizing such errors. This reveals a critical limitation of existing evaluation metrics, which fail to capture fine-grained linguistic quality, particularly morphological and grammatical agreement violations inherent to Arabic. Such overestimation poses a serious concern in educational contexts, where linguistically flawed captions could be incorrectly validated as correct references, potentially misleading learners and undermining the reliability of automated evaluation systems.

5.3.1. Comparative Image Captioning Performance Across Arabic and English

For comparative purposes, we evaluated both Gemma and LLaMA models on the English and Arabic versions of the Flickr8k dataset using 400 samples. The results are reported in Figure 5.

Overall, both models consistently achieve higher scores in English than in Arabic across nearly all evaluation metrics. This indicates that the models generate more accurate and reliable captions in English compared to Arabic.

For the LLM-based metrics, the performance is relatively comparable across languages, with higher CLAIR scores in Arabic and higher FLEUR scores in English. This suggests that the Arabic-generated captions are more closely aligned with the reference texts, while the English captions

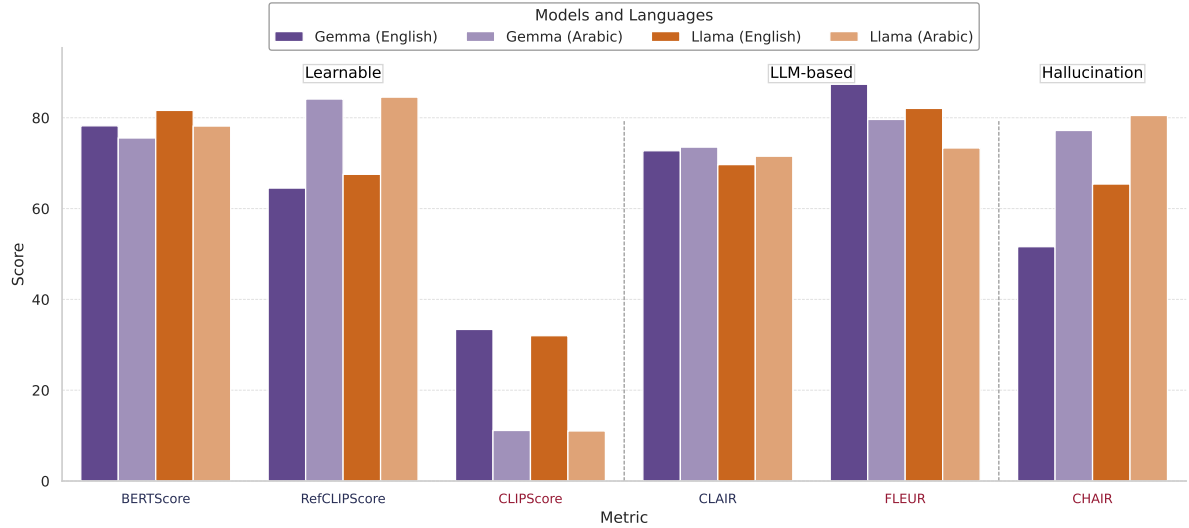


Figure 5: Performances of Gemma and LLaMA, across six metrics, on 400 samples from both Flickr8k versions : English and Arabic.

demonstrate stronger visual grounding.

A notable difference appears in the CHAIR scores across languages, with English consistently achieving better scores. This suggests that both models are more prone to hallucinations when generating captions in Arabic than in English. However, these results should be interpreted with caution. CHAIR relies on object lists derived from the MS-COCO dataset, which is predominantly English-centric and lacks adequate Arabic lexical coverage. As discussed earlier, this renders CHAIR a poorly suited metric for Arabic caption evaluation, and its scores in this language cannot be considered fully reliable.

6. Conclusion

We evaluated Multimodal Large Language Models for Arabic image captioning using the Arabic Flickr8k and JEEM datasets and nine metrics. Gemma and Gemini models achieved the strongest performance across standard metrics, demonstrating robustness in this domain. However, while all models effectively captured core semantic content, they consistently struggled with exact lexical alignment to reference captions — a limitation largely attributable to Arabic’s rich morphology and syntactic complexity.

Moreover, the tendency of the models to generate overly detailed or speculative captions highlights the persistent challenge of hallucination in generative MLLMs. While the evaluation is confined to two datasets, the results provide a significant baseline for evaluating Arabic vision-language models and offer valuable insights into their current strengths and weaknesses.

Future research will explore prompt engineering strategies grounded in the linguistic competencies framework — targeting morphological, syntactic, and lexical dimensions of Arabic — alongside metrics that better reflect true caption quality. We also plan to address hallucination and integrate linguistically-aware captioning systems into educational tools to foster richer, more accessible learning experiences.

7. Acknowledgements

The authors gratefully acknowledge FANAR.ai for granting access to their Vision Model API. This work is supported by the Directorate-General for Scientific Research and Technological Development (DGRSDT) - Algeria, and performed under the PRFU Project: C00L07N030120220002.

Imad Afyouni, Imtinan Azhar, and Ashraf Elnagar. 2021. [Aracap: A hybrid deep learning architecture for arabic image captioning](#). *Procedia Computer Science*, 189:382–389.

Rasha Saleh Al-Malki and Arwa Yousuf Al-Aama. 2023. [Arabic captioning for images of clothing using deep learning](#). *Sensors*, 23(8):3783.

Ashwaq Alsayed, Thamir M. Qadah, and Muhammad Arif. 2023. [A performance analysis of transformer-based deep learning models for arabic image captioning](#). *Journal of King Saud University - Computer and Information Sciences*, 35(9):101750.

- Satanjeev Banerjee and Alon Lavie. 2005. [Me-teor: 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*, page 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Ahlam Bashiti, Alaa Aljabari, Hadi Khaled Hamoud, Md. Rafiul Biswas, Bilal Mohammed Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouni. 2025. [Imageeval 2025: The first arabic image captioning shared task](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, page 376–389, Suzhou, China. Association for Computational Linguistics.
- David M. Chan, Suzanne Petryk, Joseph E. Gonzalez, Trevor Darrell, and John Canny. 2023. [Clair: Evaluating image captions with large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 13638–13646, Singapore. Association for Computational Linguistics.
- Xiaofu Chen, Israfel Salazar, and Yova Kementchedjieva. 2025. Specs: Specificity-enhanced clip-score for long image caption evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9406–9418.
- Khaled Bachir Delassi, Lakhdar Zeggane, Hadda Cherroun, Abdelhamid Haouhat, and Kaoutar Bouzouad. 2025. [Vqa support to arabic language learning educational tool](#).
- Yashwant Dongare, Bhalchandra M. Hardas, Rashmita Srinivasan, Vidula Meshram, Mithun G. Aush, and Atul Kulkarni. 2024. [Deep neural networks for automated image captioning to improve accessibility for visually impaired users](#). *International Journal of Intelligent Systems and Applications in Engineering*, 12(2s):267–281.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. [The Llama 3 Herd of models](#).
- Passant Elchafei and Amany Fashwan. 2025. [Multimodal arabic captioning with interpretable visual concept integration](#). (arXiv:2510.03295). ArXiv:2510.03295 [cs].
- Obeida ElJundi, Mohamad Dhaybi, Kotaiba Mokadam, Hazem Hajj, and Daniel Asmar. 2020. [Resources and end-to-end neural network models for arabic image captioning](#). In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, page 233–241, Valletta, Malta. SCITEPRESS - Science and Technology Publications.
- Jonathan Emami, Pierre Nugues, Ashraf Elnagar, and Imad Afyouni. 2022. [Arabic image captioning using pre-training of deep bidirectional transformers](#). In *Proceedings of the 15th International Conference on Natural Language Generation*, page 40–51, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.
- FutureBee AI. 2025. Arabic image caption dataset. <https://www.futurebeeai.com/dataset/multi-modal-dataset/arabic-image-caption-dataset>. Accessed: February 7, 2026.
- Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.
- Abdelhamid Haouhat, Slimane Bellaouar, Attia Nehar, Hadda Cherroun, and Ahmed Abdelali. 2025. Arabic multimodal machine learning: Datasets, applications, approaches, and challenges. *arXiv e-prints*, pages arXiv–2508.
- Mohammad Nehal Hasnine, Brendan Flanagan, Gokhan Akcapinar, Hiroaki Ogata, Kousuke Mouri, and Noriko Uosaki. 2019. [Vocabulary Learning Support System Based on Automatic Image Captioning Technology](#), volume 11587 of *Lecture Notes in Computer Science*, page 346–358. Springer International Publishing, Cham.
- Hani Hejazi and Khaled Shaalan. 2021. [Deep learning for arabic image captioning: A comparative study of main factors and preprocessing recommendations](#). *International Journal of Advanced Computer Science and Applications*, 12(11).
- Khalil Hennara, Muhammad Hreden, Mohamed Motaism Hamed, Zeina Aldallal, Sara Chrouf, and Safwan AlModhayan. 2025. [Mu-tarjim: Advancing bidirectional arabic-english translation with a small language model](#).
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Roman Le Bras, and Yejin Choi. 2022. [Clipscore: A reference-free evaluation metric for image captioning](#). (arXiv:2104.08718). ArXiv:2104.08718 [cs].

- Ming Jiang, Qiuyuan Huang, Lei Zhang, Xin Wang, Pengchuan Zhang, Zhe Gan, Jana Diesner, and Jianfeng Gao. 2019. [Tiger: Text-to-image grounding for image caption evaluation](#). (arXiv:1909.02050). ArXiv:1909.02050 [cs].
- Romrawin JP. 2025. Multilingual coco dataset. <https://huggingface.co/datasets/romrawinjp/multilingual-coco>. Accessed: February 7, 2026.
- Karima Kadaoui, Hanin Atwany, Hamdan Al-Ali, Abdelrahman Mohamed, Ali Mekky, Sergei Tilga, Natalia Fedorova, Ekaterina Artemova, Hanan Aldarmaki, and Yova Kementchedjieva. 2025. [Jeem: Vision-language understanding in four arabic dialects](#). (arXiv:2503.21910). ArXiv:2503.21910 [cs].
- Yebin Lee, Imseong Park, and Myungjoo Kang. 2024. [Fleur: An explainable reference-free evaluation metric for image captioning using a large multimodal model](#). (arXiv:2406.06004). ArXiv:2406.06004 [cs].
- Xianhang Li, Haoqin Tu, Mude Hui, Zeyu Wang, Bingchen Zhao, Junfei Xiao, Sucheng Ren, Jieru Mei, Qing Liu, Huangjie Zheng, et al. 2024. What if we recaption billions of web images with llama-3? *arXiv preprint arXiv:2406.08478*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311, Philadelphia, Pennsylvania. Association for Computational Linguistics.
- Suzanne Petryk, David M. Chan, Anish Kachinthaya, Haodi Zou, John Canny, Joseph E. Gonzalez, and Trevor Darrell. 2024. [Aloha: A new measure for hallucination in captioning models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, page 342–357, Mexico City, Mexico. Association for Computational Linguistics.
- Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2024. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2019. [Object hallucination in image captioning](#). (arXiv:1809.02156). ArXiv:1809.02156 [cs].
- Sara Sarto, Marcella Cornia, Rita Cucchiara, et al. 2025. Image captioning evaluation in the age of multimodal llms: Challenges and future perspectives. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*.
- Fanar Team, Umam Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, Mus’ab Husaini, Soon-Gyo Jung, Ji Kim Lucas, Walid Magdy, Safa Messaoud, Abubakr Mohamed, Tasnim Mohiuddin, Basel Mousi, Hamdy Mubarak, Ahmad Musleh, Zan Naeem, Mourad Ouzzani, Dorde Popovic, Amin Sadeghi, Husrev Taha Sencar, Mohammed Shinoy, Omar Sinan, Yifan Zhang, Ahmed Ali, Yassine El Kheir, Xiaosong Ma, and Chaoyi Ruan. 2025a. [Fanar: An arabic-centric multimodal generative ai platform](#).
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025b. [Gemma 3 Technical Report](#).
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). (arXiv:1904.09675). ArXiv:1904.09675 [cs].
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.
- Amir Zur, Elisa Kreiss, Karel D’Oosterlinck, Christopher Potts, and Atticus Geiger. 2024. Updating clip to prefer descriptions over captions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20178–20187.

AlignAR: Generative Sentence Alignment for Arabic-English Parallel Corpora of Legal and Literary Texts

Baorong Huang, Ali Asiri*

School of Foreign Languages, Huaihua University
Alith University College, Umm al-Qura University
Huaihua, China,
Makkah, Saudi Arabia

huangbaorong2021@gmail.com, amaasiri@uqu.edu.sa

Abstract

High-quality parallel corpora serve as the fundamental backbone for advancements in Machine Translation (MT) research and the development of effective translation pedagogy. Despite this need, robust resources for the Arabic-English language pair remain significantly scarce. Furthermore, existing datasets are often limited by their reliance on simplistic one-to-one sentence mappings, which fail to capture the structural complexities inherent in natural language translation. To address this deficiency, this paper presents AlignAR, a novel generative sentence alignment method, alongside a comprehensive new Arabic-English dataset that juxtaposes simple legal documents with complex literary texts. Our evaluation demonstrates that "Easy" datasets lack the discriminatory power to fully assess alignment methods. By reducing one-to-one mappings within our "Hard" subset, we exposed the limitations of traditional alignment techniques when faced with structural divergence. In contrast, Large Language Model (LLM) based approaches demonstrated superior robustness and adaptability. Specifically, the proposed LLM-based approaches demonstrated better robustness, achieving an overall F1-score of 85.5%, a nearly 9% improvement over previous methods. This study underscores the importance of complex benchmarks and validates the efficacy of generative models in handling the intricacies of bitext alignment. The codes and datasets are available on [GitHub](#).

Keywords: Parallel Corpora, Generative Sentence Alignment, Arabic-English Translation

1. Introduction

Parallel sentence alignment is an important step in preparing parallel data for various Natural Language Processing (NLP) applications, particularly machine translation. It involves identifying corresponding sentences between two texts that are mutual translations, as shown in Figure 1. Given that manual alignment is resource-intensive and time-consuming, various automatic methods are proposed to address the challenges.

١١: فهم حي بن يقظان ذلك كله ولم ين فيه شيئاً على خلاف ما شاهده في مكانه الكريم.	t1: Hayy understood all this and found none of it in contradiction with what he had seen for himself from his supernal vantage point.
١٢: ويقول "إن الناس لو فهموا الأمر على حقيقته لأمرهم من هذه النوازل وأقبلوا على الحق واستخروا عن هذا كله ولم يكن لأحد اعتراض بما لم يسل عن ركنه أو تفتق الأيدي على سرقته أو تنهب النفوس على أخذ معاجزه.	t2.1: If people understood things as they really are, Hayy said, they would forget these inanities and seek the Truth. t2.2: They would not need all these laws. t2.3: No one would have any property of his own to be demanded as charity or for which human beings might struggle and risk amputation.
١٣.١: وكان رأيته هو ألا يتناول أحد شيئاً إلا ما يقيم به الرقيم، وأما الأموال فلم تكن لها عهده معني. ١٣.٢: وكان يرى ما في الشرع من الأخكام في أمر الأموال كالكراهة ونسبها والبيع والربا والحدود والعقوبات، فكان يستغرب ذلك كله ويراء نظوياً.	t3.1: Hayy's own idea was that no one should eat the least bit more than would keep him on the brink of survival. t3.2: Property meant nothing to him, and when he saw all the provisions of the Law to do with money, such as the regulations regarding the collection and distribution of welfare or those regulating sales and interest, with all their statutory and discretionary penalties, he was dumbfounded. All this seemed superfluous.

Figure 1: Examples of 1-to-1, 1-to-many and many-to-many alignments between source and target sentences written in Arabic and English, respectively.

Over the past three decades, the research has

Corresponding author: amaasiri@uqu.edu.sa

progressed from early statistical and length-based models (Brown et al., 1991; Gale and Church, 1993) to heuristic, structural, and modern neural embedding-based approaches capable of handling diverse domains, combining multilingual sentence embeddings, anchor-point detection, and dynamic alignment algorithms, such as dictionary-based extraction (Althobaiti, 2022), divide and conquer (Steingrímsson et al., 2023; Zhang, 2022). More recent work explores embedding-based methods, such as overlapping fixed-length segmentation (Wang et al., 2024), multilingual embeddings (Thompson and Koehn, 2019; Liu and Zhu, 2023), contextualized document-aware sentence embeddings (Molfese et al., 2024), and particle swarm optimization (Shang and Li, 2024). Despite these advances, research on sentence alignment using Large Language Models (LLMs) remains notably scarce.

Previous alignment methods are evaluated on web datasets in English-German texts (Koehn et al., 2018) or some low-resource languages, such as Pashto-English (Koehn et al., 2020) or Estonian-Lithuanian (Sloto et al., 2023). In addition, some gold alignment datasets are released to evaluate the alignment methods for French-German pairs such as Text+Berg, Chinese-English pairs (Liu and Zhu, 2023), English-Icelandic pairs (Steingrímsson et al.,

2023), or various European languages(Molfese et al., 2024). However, the resources on English-Arabic pairs are underexplored.

To address these limitations, we conduct an in-depth investigation into the integration of Large Language Models (LLMs) for automated sentence alignment and put forward the following two main contributions:

- We introduce a novel LLM-based alignment method. This method leverages the contextual understanding of LLMs to generate high-precision initial alignments, which we rigorously evaluate against our specialized parallel corpus;
- We present the Arabic-English Parallel Corpus, a new gold-standard dataset designed to test alignment robustness across varying levels of difficulty. The corpus is categorized into a legal sub-corpus, representing structured technical prose, and a literary sub-corpus, containing complex, non-linear narratives that pose a significant challenge to previous alignment methods.

Ultimately, we hope that this study provides both a scalable methodology and a high-fidelity resource to stimulate further research into specialized bilingual alignment, particularly for low-resource and linguistically complex pairs.

2. Existing methods

2.1. BleuAlign

Sennrich and Volk (2010) proposed Bleualign, in which high-scoring sentence pairs are treated as anchor points, while dynamic programming is employed to infer optimal alignments between anchors, allowing for one-to-one and one-to-many mappings. The approach also integrates traditional length-based heuristics as a fallback mechanism in difficult cases.

2.2. VecAlign

Thompson and Koehn (2019) introduced Vecalign, which represents sentences from both languages in a shared semantic vector space using pre-trained multilingual encoders and computes cosine similarity to identify semantically corresponding sentence pairs. To ensure scalability, the authors proposed a linear-time dynamic programming algorithm that aligns sentences monotonically while permitting sentence splits and merges.

2.3. BertAlign

Liu and Zhu (2023) proposed Bertalign, in which sentences are encoded using transformer embed-

dings, enabling more precise similarity estimation than static word or sentence embeddings. Bertalign computes sentence-level similarity scores and applies dynamic programming to determine optimal alignments under monotonicity constraints.

3. Datasets

The data used in this study are divided into two main categories based on the complexity of the sentence alignment task: easy-alignment data and hard-alignment data, as shown in Table 1. The hard part consists of 5 documents, totaling 378 Arabic sentences and 774 English sentences, with the smallest source/target ratio around 0.45, which means that one Arabic sentence aligns to at least two English sentences on average. The easy part also consists of 5 documents manually aligned from Arabic laws and its English translations, totaling 892 source lines and 1093 target lines, with the smallest source/target ratio around 0.74, much higher than the hard part.

The easy-alignment subset comprises modern legal texts characterized by high structural parallelism and terminological consistency. These data were collected from the Saudi National Center for Archives and Records (NCAR), which provides official Saudi laws alongside their English translations, including the Companies Law, Anti-Bribery Law, and Anti-Commercial Concealment Law. Due to their formulaic drafting style, controlled vocabulary, and close translation correspondence, these legal texts represent low-complexity alignment data.

In contrast, the data of the hard-alignment subset are drawn from classical and literary Arabic texts, which present significant challenges due to archaic language, stylistic density, and non-literal translation strategies. The raw data are from the open-source platform Rasaifa, including the philosophical narrative Ḥayyibn Yaqzān, and short stories included in Hassan Ghazala’s A Textbook of Literary Translation (2013).

4. Method

In this study, we frame document alignment as a zero-shot inference task, prompting the models to identify correspondences based on semantic equivalence rather than lexical overlap. We propose a novel sentence alignment that considers alignment as a translation mapping task, and request the LLM to identify the translations of the source text, rather than directly performing the alignment.

Table 1: Statistics of the Arabic–English Parallel Corpus

Dataset	#AR-SENT	#AR-TKN	#EN-SENT	#EN-TKN	SENT. (%)	1-1 (%)
Easy 1	153	4,405	206	7,783	74.27	78.15
Easy 2	202	6,437	257	11,438	78.60	77.78
Easy 3	161	4,726	198	8,471	81.31	81.76
Easy 4	202	4,893	233	8,766	87.73	85.20
Easy 5	174	5,770	199	10,650	87.44	87.21
Hard 1	93	3,880	202	6,318	46.03	28.77
Hard 2	100	3,998	194	6,823	51.55	40.26
Hard 3	93	3,111	175	5,849	53.14	45.78
Hard 4	92	3,068	203	5,800	45.32	40.70
Hard 5	101	2,759	223	6,906	45.29	31.18

Note: #AR-SENT and #AR-TKN indicate sentence count and token count in the Arabic text; #EN-SENT and #EN-TKN indicate sentence count and token count in the English text; SENT(%) means the ratio of Arabic sentences to English sentences in the dataset; 1-1 (%) means the frequency (percentage) of 1-to-1 alignments in each dataset. Both source and target texts have been tokenized with Stanza before counting tokens.

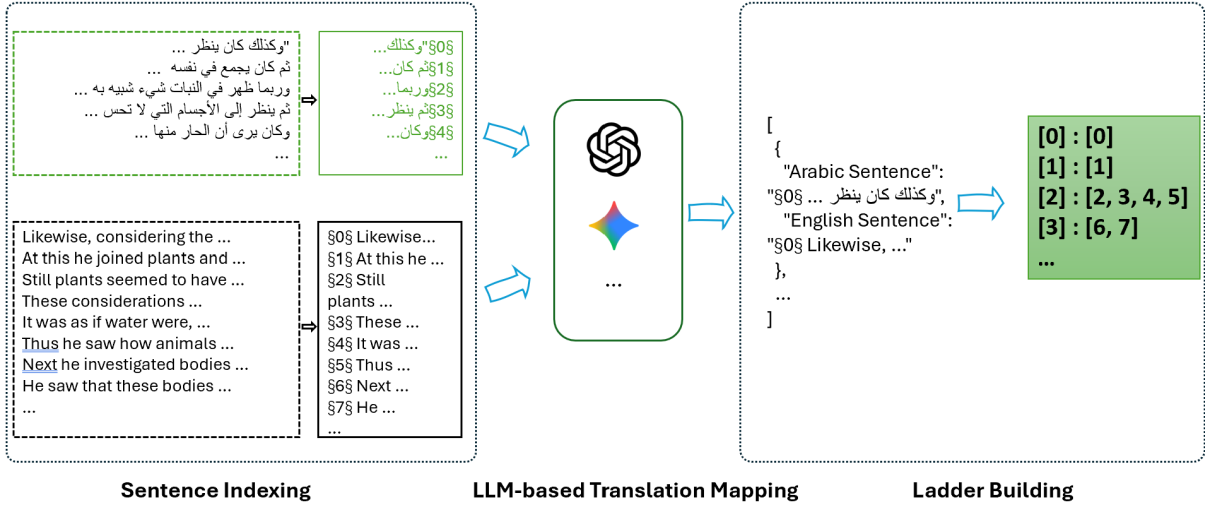


Figure 2: Our alignment procedure consists of three steps: (1) sentence indexing, i.e., explicitly indexing the lines with sequence number; (2) LLM-based translation mapping, i.e., requesting the selected LLM to select the translations from the target lines; (3) ladder building, extracting the index number from the formatted JSON output and obtaining ladder files for evaluation.

4.1. Alignment Procedure

We propose a three-step alignment procedure. In the first step, we add a sentence index (sequence number) as $[0 s_0, \dots, n s_m]$ and $[0 t_0, \dots, n t_n]$ to each source line and each target line. Furthermore, to further reduce the inadvertent addition or deletion of words by an LLM, we change the parallel aligning task into a translation mapping task, requesting the LLM to pick out the corresponding translations of the source lines from the target lines and output the entire mappings in a JSON format, together with their sentence indexes. This design addresses the issue that directing the LLM to output only sentence indices considerably degrades alignment quality. To mitigate this, we re-

quire the LLM to reproduce the original text alongside the indices. In the final step, we only extract the sentence indexes to obtain a ladder file that only contains line number mappings for evaluation, as shown in Figure 2. In this way, we obtained the precise sentence indexes required for alignment, while avoiding the issues in mapping the texts generated by LLM to the original texts, which is complicated because sometimes LLM failed to generate the texts identical with the input. For the prompt used, please see Appendix 10.1.

4.2. Evaluation Metrics

We use the strict precision, strict recall and strict F1 as our evaluation metrics. Strict true positive, as

Table 2: Comparison across different methods on easy document alignment

Item	Metric	Baselines			Ours GPT	Gemini
		BERT	BLEU	VEC		
Easy 1	P	0.936	0.966	0.961	0.987	0.987
	R	0.961	0.947	0.961	0.993	0.993
	F_1	0.948	0.957	0.961	0.990	0.990
Easy 2	P	0.985	0.919	0.985	0.990	0.980
	R	0.980	0.914	0.985	0.995	0.990
	F_1	0.982	0.916	0.985	0.993	0.985
Easy 3	P	0.981	0.925	0.975	1.000	1.000
	R	0.981	0.925	0.956	1.000	1.000
	F_1	0.981	0.925	0.965	1.000	1.000
Easy 4	P	1.000	0.944	0.970	0.980	0.990
	R	1.000	0.954	0.965	0.990	0.995
	F_1	1.000	0.949	0.967	0.985	0.992
Easy 5	P	1.000	0.965	0.994	0.989	1.000
	R	1.000	0.965	0.988	0.994	1.000
	F_1	1.000	0.965	0.991	0.991	1.000
Overall	P	0.982	0.943	0.977	0.989	0.991
	R	0.985	0.943	0.972	0.994	0.995
	F_1	0.984	0.942	0.974	0.992	0.993

Note: BERT means BertAlign, BLEU means BlueAlign and VEC means VecAlign. Bold indicates the best-performing method.

defined by Sennrich and Volk (2010), means the alignment hypothesis should be identical to the reference alignment. We adopt the strict metrics because the actual usage of the alignments in translation studies or MT training requires the precise and exact mappings between the source lines and target lines. In this case, the overlapping alignment as lax measures allowed will be considered incorrect. We take the Micro-Averaged Metrics for computing the overall scores for the easy and hard parts.

5. Experiments & Results

5.1. Experimental settings

We evaluate the alignment accuracy on the Arabic-English datasets using our proposed LLM-based alignment methods, in comparison with the previous methods, including BertAlign, BleuAlign, and VecAlign. Our experiments are conducted on a computer with Intel Ultra 7 265KF, 64GB RAM, and NVIDIA RTX 5070 Ti. Two LLMs are used in the experiments, including GPT-5.1-mini and Gemini-2.5-flash.

5.2. Experiments on legal text (easy part)

An analysis of the experimental results in Table 2 suggests that the "Easy" subsets of the dataset

may lack the discriminatory power required to fully evaluate the capabilities of aligning methods. As shown by the "Overall" metrics, Gemini achieves a near-perfect F_1 score of 0.993; however, the consistently high scores across all methodologies, all exceeding the 0.90 threshold, indicate that these specific benchmarks have reached saturation. Consequently, more challenging data are required to reveal the latent performance gaps.

Table 3: Comparison across different methods on hard document alignment

Item	Metric	Baselines			Ours GPT	Gemini
		BERT	BLEU	VEC		
Hard 1	P	0.667	0.507	0.670	0.684	0.811
	R	0.757	0.479	0.735	0.783	0.880
	F_1	0.726	0.493	0.701	0.730	0.844
Hard 2	P	0.724	0.563	0.791	0.718	0.871
	R	0.807	0.519	0.818	0.841	0.920
	F_1	0.763	0.540	0.804	0.775	0.895
Hard 3	P	0.606	0.410	0.645	0.612	0.688
	R	0.648	0.410	0.682	0.682	0.750
	F_1	0.626	0.410	0.663	0.645	0.717
Hard 4	P	0.778	0.618	0.826	0.884	0.903
	R	0.865	0.547	0.854	0.944	0.944
	F_1	0.819	0.580	0.840	0.913	0.923
Hard 5	P	0.699	0.340	0.780	0.748	0.880
	R	0.742	0.310	0.804	0.794	0.907
	F_1	0.720	0.324	0.792	0.770	0.893
Overall	P	0.696	0.494	0.743	0.729	0.831
	R	0.757	0.462	0.780	0.809	0.831
	F_1	0.726	0.477	0.761	0.767	0.855

5.3. Experiments on literary text (hard part)

The results on the challenging document alignment task (Table 3) provide a much clearer differentiation of model capabilities. As the task complexity increases, the "ceiling effect" observed in the easy datasets disappears, revealing significant discrepancies in model robustness. Gemini achieves an overall F_1 score of 0.855, which represents a substantial margin over the BertAlign baseline (0.726) and the GPT-based approach (0.767). The sharp decline in performance across all baseline methods in "Hard" datasets underscores their limitations. For instance, BleuAlign's overall F_1 plummeted from 0.942 to 0.477, and BertAlign's overall F_1 dropped from 0.984 to 0.726. More importantly, the divergence between GPT and Gemini on the hard datasets—most notably in Hard 5, where Gemini outperforms GPT by over 12 percentage points in F_1 —indicates that the selection of LLMs also exerts considerable influence on the alignment quality.

6. Conclusions

In this paper, we introduced a generative sentence alignment method and a new Arabic–English parallel dataset focusing on legal and literary texts. Our findings reveal the limitations of previous methods. In contrast, the LLM-based methods demonstrated better robustness, maintaining high precision and recall where other methods degraded. Beyond these empirical results, the proposed generative alignment approach provides a flexible and efficient alternative to conventional techniques, while requiring relatively modest computational resources. Consequently, our approach has the potential to support improvements in downstream machine translation (MT) systems, as well as broader applications in cross-lingual transfer and multilingual understanding. Furthermore, the released Arabic–English dataset contributes a domain-specific resource that can support future research in legal and literary translation. Future work can extend this approach to additional low-resource language pairs, and explore multi-agent or self-refinement strategies to further enhance alignment quality.

Limitations

The proposed LLM-based document alignment methods can effectively and accurately align the low-resource Arabic text to English text. However, the required time for the alignment depends on the processing time of the LLMs. In addition, the format of the responses is also essential for the successful parsing and construction of the alignment mappings. Furthermore, the length of the document is also constrained by the context window of the large language models. In our experiments, one simple prompt is used, as described in 10.1 and the effects of advanced prompt methods, such as chain-of-reason (COT), self-reflection, or agentic design, are not explored. Prior work suggests that these methods, especially those involving multi-agent collaboration, such as reflection or reviewer agent mechanisms(Bo et al., 2024) or ensemble mechanisms(Qian et al., 2026), can enhance the performance and reasoning reliability of LLMs. Investigating whether such approaches further improve alignment quality remains an important direction for future research.

In addition, it is well-known that LLMs have difficulties in handling long texts. Future research may explore how long texts, such as an entire novel with hundreds of pages and its translation, can be effectively aligned by LLMs.

7. Acknowledgments

The authors extend their appreciation to Umm Al-Qura University, Saudi Arabia for funding this research work through grant number: 26UQU4350258GSSR01.

8. Funding Statement

This research work was funded by Umm Al-Qura University, Saudi Arabia under grant number: 26UQU4350258GSSR01.

9. Bibliographical References

References

- Maha Jarallah Althobaiti. 2022. A simple yet robust algorithm for automatic extraction of parallel sentences: A case study on arabic-english wikipedia articles. *IEEE Access*, 10:401–420.
- Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. 2024. [Reflective multi-agent collaboration based on large language models](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 138595–138631. Curran Associates, Inc.
- Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 169–176.
- William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.
- Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742.
- Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *EMNLP 2018 Third Conference on Machine Translation (WMT18)*, pages 726–739. Association for Computational Linguistics.
- Lei Liu and Min Zhu. 2023. Bertalign: Improved word embedding-based sentence alignment for chinese–english parallel corpora of literary texts. *Digital Scholarship in the Humanities*, 38(2):621–634.

Francesco Maria Molfese, Andrei Stefan Bejgu, Simone Tedeschi, Simone Conia, and Roberto Navigli. 2024. Crocoalign: A cross-lingual, context-aware and fully-neural sentence alignment system for long texts. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2209–2220.

Yiyue Qian, Shinan Zhang, Yun Zhou, Haibo Ding, Diego Socolinsky, and Yi Zhang. 2026. [Collabeval: Enhancing llm-as-a-judge via multi-agent collaboration](#).

Rico Sennrich and Martin Volk. 2010. Mt-based sentence alignment for ocr-generated parallel texts. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*, pages 1–10, Denver, USA.

Wei Shang and Xin Li. 2024. Construction and alignment features of english corpus based on particle swarm optimization algorithm. In *2024 International Conference on Distributed Systems, Computer Networks and Cybersecurity (ICDSCNC)*, pages 1–5. IEEE.

Steve Sloto, Brian Thompson, Huda Khayrallah, Tobias Domhan, Thamme Gowda, and Philipp Koehn. 2023. Findings of the wmt 2023 shared task on parallel data curation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 95–102.

Steinþór Steingrímsson, Hrafn Loftsson, and Andy Way. 2023. SentAlign: Accurate and Scalable Sentence Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Singapore, Singapore. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Xiaotian Wang, Takehito Utsuro, and Masaaki Nagata. 2024. Document alignment based on overlapping fixed-length segments. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 51–61.

Wu Zhang. 2022. Improve sentence alignment by divide-and-conquer. *arXiv preprint arXiv:2201.06907*.

10. Language Resource References

N/A

Appendix

10.1. Prompt Template and Sample Responses

The following prompt template is used in the alignment process.

You are an expert in translation. Align two documents and output bilingual pairs.

Document A is in Arabic, and Document B is in English. Both documents contain sentences that are similar in meaning but may not be direct translations of each other.

Your task is to find the translations from Document B for lines in Document A and output them as bilingual pairs. Use the source line as one sentence as much as possible. In some cases, more than one source lines are matched to more than one target lines.

Find the translations for all the source lines. Do not merge the source lines unless absolutely necessary.

Produce the complete bilingual alignment in one large JSON array.

When finding the translation, ignore the line number such as §0, §1 at the beginning of each sentence and keep the line number in the output.

Document A (Arabic):

{arabic_text}

Document B (English):

{english_text}

Output Format:

[{"Arabic Sentence": "§0 نظام القضاء",
"English Sentence": "§0 Law of the Judiciary" }]

The sample response is described as follows:

["Arabic Sentence": "§0 الشركة كيان قانوني يؤسس وفقاً لأحكام النظام بناءً على عقد تأسيس أو نظام أساس يلتزم بمقتضاه §0 شخصان أو أكثر بأن يساهم كل منهم في مشروع يستهدف الربح بتقديم حصة من مال أو عمل أو منهما معاً لاقتسام ما ينشأ عن هذا المشروع من ربح أو خسارة، واستثناء من ذلك، يجوز -وفقاً لأحكام النظام- أن تؤسس الشركة بالإرادة المنفردة لشخص واحد، ويجوز تأسيس شركات غير ربحية وفقاً لما ورد في الباب (السابع) من النظام.",

"English Sentence": "§0 A company is a legal entity incorporated in accordance with the provisions of this Law pursuant to articles of incorporation or articles of association under which two or more persons undertake to participate in a for-profit enterprise by contributing property or work, or both, to share any profit realized or loss incurred from such enterprise. §1 As an exception, a company may, under this Law, be incorporated by a single person, and a non-profit company may be incorporated pursuant to the provisions of Part 7 of this Law.",

...
]

Helpful or Harmful? The Dual Role of Linguistic Features in LLM-Based Dialectal Machine Translation

Abdelhalim Hafedh Dahou¹, Mohamed Amine Cheragui²

¹GESIS – Leibniz-Institute for the Social Sciences, Cologne, Germany

²Department of Mathematics and Computer Science, University of Ahmed DRAIA, Adrar, Algeria
abdelhalim.dahou@gesis.org, m_cheragui@univ-adrar.edu.dz

Abstract

Large Language Models (LLMs) have shown promising results in dialectal machine translation, yet the impact of explicit linguistic features remains underexplored. This paper examines whether part-of-speech (POS) tags and diacritization help or hinder LLM-based translation between Algerian dialect (Darija) and Modern Standard Arabic (MSA). Using a linguistically enriched subset of the PADIC dataset, we conduct bidirectional experiments across several frontier and open-weight LLMs, evaluated with automatic metrics and human judgments of adequacy and fluency. Results reveal a dual and asymmetric effect: diacritics can improve adequacy in the MSA → Algerian dialect direction, while POS tags and forced diacritization often introduce noise, especially for Algerian dialect → MSA translation. We further observe a mismatch between traditional overlap-based metrics and human evaluation, suggesting limitations in current evaluation practices. Overall, explicit linguistic augmentation does not consistently benefit LLM-based dialectal translation and must be applied cautiously. The code, prompts, and datasets are available at: [FeatureDual-MT Repository](#)

Keywords: LLM, Machine Translation, POS, Diacritization, Algerian dialect, MSA

1. Introduction

Arabic natural language processing poses significant challenges due to the language’s rich morphology, flexible word order, and high degree of lexical ambiguity (Habash, 2010). Two linguistic factors are particularly important for addressing these challenges: Part-of-Speech (POS) information and diacritization. POS tagging provides explicit syntactic cues by identifying the grammatical role of words (Petrov et al., 2012), while diacritics encode short vowels and morphological markers that are typically omitted in written Arabic, resulting in substantial lexical and syntactic ambiguity (Habash and Rambow, 2007; Zitouni et al., 2006). In the absence of these cues, a single surface form may correspond to multiple grammatical categories and meanings, complicating automatic language understanding. Consequently, both POS tagging and diacritization have been shown to be effective for improving disambiguation and downstream Arabic NLP tasks, including machine translation (Diab and Hacıoglu, 2004; Habash, 2013).

In machine translation, linguistic features such as POS tags and diacritization remain valuable for morphologically rich languages like Arabic. POS information helps models better capture syntactic dependencies, while diacritization reduces ambiguity by providing missing morphological information (Habash, 2019; Zalmout and Habash, 2020). Prior work has shown that incorporating linguistic annotations can complement end-to-end neural models, improving generalization and robustness, particularly in low-resource and dialectal settings (Sennrich and Haddow, 2016; Currey and

Heafield, 2020). For Arabic, combining POS-aware and diacritic-sensitive representations has been shown to enhance disambiguation and translation quality (Abdul-Mageed and Bouamor, 2021; Khalifa et al., 2022). Even with transformer-based and large language model (LLM) architectures, explicit linguistic enrichment can improve handling of complex morphology and closely related language varieties (Edman and Sogaard, 2023; Liu and Neubig, 2024).

In this work, we study the impact of incorporating POS tagging and diacritization as auxiliary linguistic features in bidirectional machine translation between Algerian dialect (Darija) and Modern Standard Arabic (MSA) using LLM-based systems. We conduct a systematic evaluation of POS- and diacritic-aware translation settings in both directions, analyzing their effect on ambiguity reduction and translation quality. Our results shed light on the effect of explicit syntactic and morphological cues in translation accuracy and linguistic consistency for dialectal Arabic.

2. Linguistic Differences Between Algerian Dialect and Modern Standard Arabic

MSA serves as the standardized variety used in formal, written, and institutional contexts, whereas Algerian dialect (*Darja*) is the primary medium of everyday communication. Despite their shared historical origin, the two varieties differ substantially at the phonological, morphological, syntactic, and lexical levels. Algerian dialect is characterized by

reduced inflectional morphology, distinct syntactic patterns, extensive lexical borrowing, and the absence of a standardized orthography. These structural differences result in limited mutual intelligibility in spoken contexts and have important implications for linguistic analysis and language technology.

2.1. Phonological Differences

Algerian dialect exhibits significant phonological divergence from MSA. Vowel reduction and elision are common, resulting in consonant clusters that are rare in MSA. Additionally, several consonants undergo systematic variation. The MSA phoneme /q/ is frequently realized as (Holes, 2004; Kouloughli, 1999):

- the palatal sound [g / ث] is used in several regional varieties. It is attested in northern cities such as Annaba and Sétif, and is especially prevalent in Bedouin dialects. This sound is also widely employed in southern regions of Algeria, including Adrar, Bechar, and Tamanrasset, where it constitutes a stable phonological feature.
- The glottal stop [ʔ / ء] is used in the Algerian dialect spoken in Tlemcen.
- The post-palatal sound [k / ك] represents a distinctive feature of the Algerian dialect and is not attested in other North African dialects. It is primarily used in rural varieties and is also found in several regions and cities, including Kabylia, Jijel, Msirda, and Trara.

Overall, the letter "ق" is not the only case of phonetic variation in Algerian dialect. Other letters, such as: "ج", "غ", and "ث" also display different pronunciations across regions, further illustrating the phonological diversity of the Algerian dialect in comparison with MSA.

2.2. Morphological Differences

Although both varieties share a root-and-pattern morphological system, Algerian dialect shows extensive morphological simplification. Case endings, dual forms, and feminine plural which are obligatory in MSA, are absent in Algerian dialect (Holes, 2004), the first and the second person of the singular form are conjugated in the same way in the Algerian dialect. Verbal paradigms are reduced, and tense–aspect distinctions are often expressed analytically rather than through inflection (Sayahi, 2014). Negation provides another clear contrast, as Algerian dialect typically uses a bipartite negation construction (*ma*–...–*š* / ما...ش), unlike the

single negation particle used in MSA (Kouloughli, 1999).

- **MSA:** لا أكتب — *lā aktubu*
- **Algerian dialect:** ما نكتبش — *ma n-ktb-š*

2.3. Syntactic Differences

Syntactically, Algerian dialect tends to favor Subject–Verb–Object (SVO) word order, whereas MSA allows both Verb–Subject–Object (VSO) and SVO structures (Versteegh, 2014). Algerian dialect also makes frequent use of resumptive pronouns in relative clauses and relies on discourse particles rather than inflectional markers (Sayahi, 2014). Another important distinction between MSA and Algerian dialect concerns verb–subject agreement and how it is affected by word order. In MSA, verb agreement varies depending on whether the verb comes before or after the subject: it can be either partial or full. In contrast, dialectal Arabic consistently uses full agreement, no matter where the verb appears in the sentence.

Example:

In MSA:

- وصل الطلاب إلى المدرسة
waáala al-áullābu ilā al-madrasah
“The students arrived at the school.”
In this case, the verb appears in the singular form despite the plural subject, illustrating partial agreement.
- الطلاب وصلوا إلى المدرسة
al-áullābu waáalū ilā al-madrasah
“The students arrived at the school.”
Here, the verb fully agrees with the plural subject.

In Algerian dialect:

- الطلاب وصلو للمدرسة
“The students arrived at the school.”
The verb shows full agreement with the subject regardless of word order.

2.4. Lexical Differences

Lexical variation represents one of the most salient differences between Algerian dialect and MSA. Algerian dialect incorporates numerous loanwords from Berber, French, Turkish, Spanish, and Italian as a result of historical contact (Benrabah, 2014) (see Table 1). These borrowings are often integrated into Arabic morphological patterns and may co-occur with Arabic vocabulary within the same utterance (Bouhadiba, 2017).

Loanword	Source	English
كسكسي	Berber <i>kseksu</i>	couscous
كوزينة	French <i>cuisine</i>	kitchen
باي	Turkish <i>bey</i>	governor
ساباط	Spanish <i>zapato</i>	shoe
بانيو	Italian <i>bagno</i>	bathroom

Table 1: Frequent loanwords in Algerian dialect by source language.

2.5. Orthography and Standardization

Unlike MSA, which has a standardized orthography, Algerian dialect lacks an official writing system (see Table 2) and is primarily oral. When written, particularly in digital contexts, it may appear in Arabic script, Latin script (Arabizi), or mixed forms, resulting in significant spelling variation (Bassiouny, 2009).

Algerian dialect	MSA	English
درك	الآن	now
dark		
دروك		
drouk		

Table 2: Example of Algerian dialect variants and Arabizi forms corresponding to MSA.

3. Literature review

3.1. Machine Translation in the Era of LLMs

Recent work has shown a clear shift from traditional supervised neural machine translation (NMT) toward LLMs for Dialectal Arabic (DA) to MSA translation. (Elneima et al., 2024) introduced a large-scale benchmark for DA–MSA translation covering five major dialect groups and demonstrated that zero-shot and few-shot prompting with ChatGPT outperforms fine-tuned NMT systems, establishing strong baselines for LLM-based approaches. Similarly, (Atwany et al., 2024) compared supervised Transformer models with prompting-based LLMs and found that few-shot prompting with GPT-3.5 substantially surpasses fine-tuned models trained on MADAR (Bouamor et al., 2018), highlighting the difficulty of dialect modeling using conventional supervision alone.

Several studies explored combining LLMs with data-centric approaches. (Abdelaziz et al., 2024) leveraged ChatGPT-generated translations to construct large-scale parallel corpora, showing that training higher-capacity Transformer models on mixed synthetic and human-curated data significantly improves DA–MSA translation quality.

In a similar vein, (Khered et al., 2025) introduced Dial2MSA-Verified, a high-quality multi-dialect dataset with human verification and multiple references, and demonstrated that AraT5 achieves strong performance across dialects, particularly under joint multi-dialect training.

Other work examined efficiency and modeling trade-offs. (Alabdullah et al., 2025) evaluated both training-free prompting and resource-efficient fine-tuning across multiple LLMs, showing that few-shot prompting consistently outperforms zero-shot strategies, while carefully scaled fine-tuning can exceed prompting-based systems when sufficient high-quality data is available. Focusing on Algerian dialect, (Babaali et al., 2025) compared traditional Seq2Seq models with ChatGPT prompting and showed that few-shot prompting yields superior performance in both DA–MSA and MSA–DA directions. Dialect-specific systems were also explored by (Sibae et al., 2025), who developed SHAMI-MT for Syrian Arabic using AraT5 and demonstrated high translation quality using LLM-based evaluation. Finally, (Obeidat et al., 2025) introduced Ir-bidDial, a Jordanian dialect dataset, and showed that few-shot prompting with GPT-based models consistently outperforms fine-tuned NMT systems, even when dialect-aware prompting constraints are applied.

3.2. Influence of POS Tagging and Diacritization on MT

Prior to the widespread adoption of LLMs, several studies investigated the role of linguistic preprocessing in Arabic machine translation. (Baniata et al., 2018) proposed a multitask NMT framework that jointly learns DA–MSA translation and POS tagging, demonstrating that syntactic supervision leads to consistent BLEU improvements for both Levantine and Maghrebi dialects. Complementary work examined the impact of Arabic diacritization on statistical machine translation. (Diab et al., 2007) showed that full diacritization degrades the performance of SMT due to data sparsity, while selective, linguistically motivated schemes produce comparable or slightly improved results. Similarly, (Alqah-tani et al., 2016) found that partial diacritization strategies combining lexical and inflectional cues consistently outperform both fully diacritized and undiacritized baselines. Together, these studies indicate that linguistically informed preprocessing can benefit Arabic MT, but overly aggressive annotation increases sparsity and harms performance.

4. Experimental Apparatus

4.1. Procedure

Our experimental procedure is designed to investigate how linguistic features, model capability, and translation metrics affect bidirectional MT between Algerian dialect and MSA. The evaluation framework answers the following research questions:

- We first address **RQ1** in order to measure the effects of POS and diacritics on translation quality and ambiguity resolving.
- We then address **RQ2** by evaluating the ability of LLMs to understand and generate Algerian dialect and MSA.
- Finally, to address **RQ3**, we compare automatic evaluation scores with human judgments across all settings to assess their alignment for dialectal Arabic.

4.2. Dataset

Experiments are conducted on a subset of the PADIC (Meftouh et al., 2015) dataset, which contains sentence-aligned Algerian dialect and MSA pairs and is commonly used for Arabic dialects translation. To incorporate syntactic information, POS tags are added for both language varieties. Algerian dialect POS annotations are generated using the Algerian BERT-based model proposed in (Cheragui et al., 2023), which outputs token-level POS sequences from raw Algerian input. MSA POS tagging is performed using the FARASA toolkit (Darwish and Mubarak, 2016), a state-of-the-art and widely adopted analyzer for MSA. We additionally include diacritics information as an auxiliary linguistic feature. Diacritization for both is performed manually by a linguistic expert to ensure high annotation quality and to avoid noise introduced by automatic diacritization systems. The resulting dataset contains 475 parallel sentence pairs. The average sentence length is 7.02 tokens for Algerian dialect and 8.02 tokens for MSA, revealing a consistent length asymmetry between the two varieties. As shown in introduction, this difference reflects more explicit morphological and syntactic expression in MSA and is a known source of ambiguity in dialect–standard Arabic translation.

4.3. Baseline Models

To evaluate the impact of linguistic features, we selected a set of state-of-the-art LLMs covering proprietary and open-weight models, including GPT-4o-mini, GPT-4o (Latest), and GPT-4-1106-preview from OpenAI (Achiam et al., 2023); Claude-3.5 Sonnet from Anthropic (Anthropic, 2024); Gemini 1.5

Pro from Google (Team et al., 2023); and the open-weight models LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024) and Gemma-2-9B-Instruct (Team et al., 2024). These models span a range of capacities and architectures commonly used in multilingual and translation benchmarks.

This selection enables a controlled comparison of how models with different capacities and training paradigms handle the structural variability and ambiguity characteristic of dialectal Arabic translation. Open-weight models were evaluated locally in quantized configurations using the Ollama inference framework¹, while proprietary models were accessed through their official APIs: OpenAI², Anthropic³, Google Gemini⁴. All models were assessed using the default hyperparameter settings and the following prompt format.

Translation Prompt Template

Objective: Translate a given sentence from {SOURCE_LANGUAGE} to {TARGET_LANGUAGE}.

Your input will include a sentence in {SOURCE_LANGUAGE} along with: - Diacritization for each word - Part-of-speech tags for each word

Use this information to accurately translate the sentence into {TARGET_LANGUAGE}. The translation should preserve the original meaning while adhering to {TARGET_LANGUAGE} norms.

Steps:

1. Understand Context: Analyze the sentence, diacritics, and POS tags to grasp its structure and meaning.

2. Word-by-Word Translation: Use diacritization and grammatical class to select appropriate {TARGET_LANGUAGE} equivalents.

3. Reconstruction: Form a grammatically correct and coherent {TARGET_LANGUAGE} sentence.

4. Meaning Verification: Ensure the translation accurately reflects the original intent and nuances.

Output Format: Provide ONLY one translated sentence in {TARGET_LANGUAGE}.

4.4. Metrics

We report results using four automatic evaluation metrics: COMET as a neural-based metric for similarity of vector representations calculation,

¹<https://ollama.com/>

²<https://platform.openai.com/>

³<https://platform.claude.com/>

⁴<https://ai.google.dev/gemini-api>

Table 3: Evaluating LLMs on MT with/without POS tags (ALG→MSA). Arrows indicate change relative to W/O. Model suffixes indicate release month.

Model	Comet		MARBERT-v2		BLEU		Chrf++	
	W/O	W/	W/O	W/	W/O	W/	W/O	W/
GPT-4o-mini-07	73.14	69.87↓	75.84	74.04↓	5.44	5.04↓	32.35	30.75↓
GPT-4o-Latest	78.48	77.39↓	79.11	78.34↓	7.95	7.11↓	38.41	36.37↓
GPT-4-1106	72.99	71.59↓	74.6	74.56↓	5.34	5.67↑	30.26	31.86↑
Claude-3.5-sonnet-06	78.57	76.98↓	79.69	79.09↓	8.30	8.56↑	39.28	38.78↓
Claude-3.5-sonnet-10	79.19	77.48↓	80.36	80.03↓	9.02	9.04↑	39.99	39.77↓
Claude-3-sonnet-02	67.12	67.80↑	68.19	69.92↑	3.23	3.55↑	22.80	26.05↑
Claude-opus-02	75.70	74.21↓	77.62	76.77↓	6.49	7.05↑	34.50	33.85↓
Claude-haiku-03	67.59	62.56↓	67.78	64.68↓	3.48	2.64↓	22.82	21.79↓
Gemini 1.5 Pro	76.60	79.19↑	75.63	80.36↑	6.84	9.29↑	34.52	40.34↑
Gemini Flash	69.46	66.73↓	70.76	68.20↓	5.61	4.29↓	29.70	28.68↓
Gemini Flash-8B	70.51	66.91↓	71.07	68.05↓	4.10	2.61↓	27.63	24.51↓
Llama3.1-8B	55.32	50.56↓	62.88	60.06↓	0.63	0.34↓	14.44	13.23↓
Gemma2-9B	64.36	58.63↓	68.18	65.27↓	1.83	1.46↓	20.08	17.60↓

MARBERT-v2 (Abdul-Mageed et al., 2021) (for MSA→ALG) and CamelBERT-MSA (Inoue et al., 2021) (for ALG→MSA) for contextualized semantic similarity, BLEU for n-gram precision, and ChrF++ for character-level overlap. To complement the automated metrics, the human evaluation protocol (Costa-Jussà et al., 2022) looks at fluency and adequacy to ensure the translations sound natural and stay true to the original meaning. Using a simple three-point scale (0, 0.5, 1.0), we evaluate how well models preserve meaning and produce readable translations. Two expert annotators, trained on the evaluation protocol, independently assessed the models’ outputs, achieving inter-rater agreement scores of 84.59% for fluency and 86.78% for adequacy.

5. Results and Discussion

5.1. Impact of Linguistic Features (RQ1)

Adding linguistic features was intended to help, but in practice, it often acted as a distraction rather than a helpful signal. Across all experiments, the baseline (w/o features) proved to be the most reliable, where with auxiliary features frequently failed.

- MSA → ALG (Formal to Dialect): in this direction, diacritics (DIACR) offered a signal boost according to human evaluators (Table 7). Adequacy increased for both GPT-4 (0.88 → 0.93) and Gemini 1.5 pro (0.91 → 0.95), indicating that vowel markers aid the models in decoding formal Arabic morphology before "translating" it into the dialect. POS tags, on the other hand, were a fluency disaster (0.93 → 0.31 for GPT-4). Adding more information (POS tokens) causing labels to be mixed into the sen-

tence itself, where the translations of Llama3.1 prove this.

- ALG → MSA (Dialect to Formal): results indicate a strong negative impact in this direction. Performance drops when diacritical marks are introduced into Algerian dialect, a variety that is rarely written with diacritics in real-world usage. For instance, Claude-3.5’s BLEU score drops from 9.02 to 3.39. Imposing diacritics on a naturally non-diacritized script shifts the input into a representation unfamiliar to the model, hindering its ability to align dialectal forms with their corresponding MSA version.
- Humans vs. Machines: Humans and LLMs process language differently. Humans use diacritics to reduce ambiguity, while LLMs rely on contextual representations learned from data. Explicit cues such as POS tags can override these learned contextual signals and introduce noise, especially in low-resource settings with limited annotated data.

5.2. Overall Model Performance (RQ2)

A clear distinction in model efficacy is revealed by the comparative analysis. Close-weight models perform better in both configurations, including GPT-4o-Latest and Claude 3.5 Sonnet (20241022). With a COMET score of 79.19 and a ChrF++ of 39.99 in the ALG → MSA direction, Claude 3.5 Sonnet exhibits the highest generalization performance. Significant obvious asymmetry phenomena where models continuously exhibiting higher generation quality when translating from the ALG → MSA direction. Higher BLEU scores for ALG → MSA approximate 9.0, whereas the MSA → ALG direction

Table 4: Evaluating LLMs on MT with/without POS tags (MSA→ALG). Arrows indicate change relative to W/O.

Model	Comet		MARBERT-v2		BLEU		Chrf++	
	W/O	W/	W/O	W/	W/O	W/	W/O	W/
GPT-4o-mini-07	65.88	56.57↓	85.53	86.18↑	2.94	0.18↓	25.82	18.20↓
GPT-4o-Latest	67.67	59.19↓	87.84	88.62↑	5.60	5.52↓	31.16	33.26↑
GPT-4-1106	65.25	58.05↓	84.76	85.17↑	3.30	0.38↓	28.16	20.71↓
Claude-3.5-sonnet-06	68.12	61.85↓	89.49	88.88↓	6.61	3.88↓	33.59	28.45↓
Claude-3.5-sonnet-10	67.56	62.56↓	89.28	90.08↑	7.32	4.05↓	34.11	29.12↓
Claude-3-sonnet-02	59.94	56.49↓	86.64	85.46↓	1.80	0.43↓	22.51	19.24↓
Claude-opus-02	66.55	62.23↓	87.70	87.84↑	5.25	2.33↓	29.69	25.31↓
Claude-haiku-03	62.12	53.78↓	86.65	85.60↓	1.78	0.27↓	20.63	14.95↓
Gemini 1.5 Pro	66.90	60.11↓	87.32	88.64↑	5.25	0.78↓	32.04	22.45↓
Gemini Flash	66.47	58.46↓	87.95	87.68↓	3.58	0.31↓	26.85	19.95↓
Gemini Flash-8B	66.16	57.70↓	87.54	87.11↓	1.73	0.21↓	22.98	18.49↓
Llama3.1-8B	55.13	46.63↓	81.01	76.77↓	0.22	0.05↓	13.86	10.36↓
Gemma2-9B	45.58	50.36↑	79.27	82.68↑	0.13	0.13	4.71	10.77↑

exhibits a marginal decline, peaking at 7.3. Such a disparity suggests that there is a generation bottleneck for low-resource dialects. While the models show sufficient inferential capacity to parse dialectal nuances, they lack the syntactic fluency required for high-quality generation in non-standardized variants.

Additionally, the findings show that smaller open-weights models perform much more poorly. With BLEU scores below 1.0, Llama 3.1 8b and Gemma 2 9b both demonstrate barely useful functionality. Human evaluation metrics (Table 7 and 8) support this functionality collapse, with Llama 3.1 8b reporting adequacy and fluency scores as low as 1.58, demonstrating a near total failure to maintain cross-lingual alignment for this particular pair of languages.

5.3. Metric Analysis (RQ3)

Human assessment of adequacy and fluency (Table 7) indicates an obvious less degradation, whereas automated metrics such as BLEU and ChrF++ demonstrate sharp decreases when diacritical marks are added. A notable discrepancy exists between statistical metrics and human linguistic understanding due to measuring just the overlap with reference without taking semantics into consideration. While most models suffered from performance decay when processing diacritics, Gemini 1.5-Pro and GPT-4 latest demonstrated a positive trend in human-perceived adequacy ($0.91 \rightarrow 0.94$, and $0.88 \rightarrow 0.92$).

MARBERT-v2 scores, which monitored adequacy changes more accurately than traditional MT metrics, notably reflected this beneficial effect as well. The contextual semantic modeling of Arabic in MARBERT-v2 is responsible for this behavior,

which makes it especially well-suited for automatically assessing Arabic translation. Overall, these findings show a significant evaluation gap, where improvements in lexical clarity that are easily noticeable by human evaluators are frequently missed by traditional statistical measures.

5.4. Error Analysis

To gain deeper insight into the impact of linguistic augmentation and the quality of LLMs translation, we performed a qualitative error analysis focusing on the output generated by the best performing model (Claude 3.5 sonnet) in each translation direction.

ALG to MSA Without linguistic features most adequacy errors stem from the prevalence of code-switching in Algerian dialect input. In particular, sentences often contain French lexical words (such as *باريفزول*, *بيان*, and *لانستيزي*) (for example, *good*, *anesthesia*) written in Arabic script, which frequently lead to incorrect semantic interpretation. As a result, models tend to either mistranslate these words or ignore their intended meaning, causing substantial semantic drift. Fluency errors are comparatively less frequent in this direction. However, in some cases, outputs retain untranslated Algerian lexical words that often appearing at the beginning of the sentence, indicating incomplete normalization into standard Arabic.

Incorporating linguistic enhancement introduces additional challenges. The presence of dense annotations increases input complexity and introduces noise, which can hinder model comprehension. Although outputs typically remain well-formed at the surface level, they often exhibit severe se-

Table 5: Evaluating LLMs on MT with/without Diacritics (MSA→ALG). Arrows indicate change relative to W/O.

Model	Comet		MARBERT-v2		BLEU		ChrF++	
	W/O	W/	W/O	W/	W/O	W/	W/O	W/
GPT-4o-mini-07	65.88	61.36↓	85.53	84.03↓	2.94	2.29↓	25.82	23.84↓
GPT-4o-Latest	67.67	63.58↓	87.84	87.14↓	5.6	4.62↓	31.16	28.68↓
GPT-4-1106	65.25	64.39↓	84.76	88.13↑	3.3	3.88↑	28.16	28.33↑
Claude-3.5-sonnet-06	68.12	64.48↓	89.49	89.22↓	6.61	6.77↑	33.59	31.99↓
Claude-3.5-sonnet-10	67.56	65.07↓	89.28	88.41↓	7.32	7.02↓	34.11	33.41↓
Claude-3-sonnet-02	59.94	57.74↓	86.64	86.36↓	1.8	2.18↑	22.51	20.39↓
Claude-opus-02	66.55	64.57↓	87.7	88.16↑	5.25	5.34↑	29.69	30.1↑
Claude-haiku-03	62.12	58.17↓	86.65	86.53↓	1.78	1.53↓	20.63	18.17↓
Gemini 1.5 Pro	66.9	64.68↓	87.32	88.81↑	5.25	5.52↑	32.04	30.78↓
Gemini Flash	66.47	63.07↓	87.95	88.46↑	3.58	2.74↓	26.85	26.16↓
Gemini Flash-8B	66.16	62.68↓	87.54	88.25↑	1.73	1.2↓	22.98	22.67↓
Llama3.1-8B	55.13	54.79↓	81.01	81.61↑	0.22	0.34↑	13.86	15.78↑
Gemma2-9B	45.58	47.36↑	79.27	84.12↑	0.13	0.19↑	4.71	7.73↑

Table 6: Evaluating LLMs on MT with/without Diacritics (ALG→MSA). Arrows indicate change relative to the W/O setting.

Model	Comet		Camelbert-MSA		BLEU		ChrF++	
	W/O	W/	W/O	W/	W/O	W/	W/O	W/
GPT-4o-mini-07	73.14	48.68↓	75.84	49.65↓	5.44	1.08↓	32.35	9.77↓
GPT-4o-Latest	78.48	54.65↓	79.11	49.78↓	7.95	2.32↓	38.41	15.33↓
GPT-4-1106	72.99	53.61↓	74.6	50.27↓	5.34	1.75↓	30.26	13.9↓
Claude-3.5-sonnet-06	78.57	56.23↓	79.68	53.24↓	8.3	3.59↓	39.28	17.83↓
Claude-3.5-sonnet-10	79.19	56.08↓	80.36	53.42↓	9.02	3.39↓	39.99	17.45↓
Claude-3-sonnet-02	67.12	43.12↓	68.19	45.59↓	3.23	0.51↓	22.8	5↓
Claude-opus-02	75.7	55.54↓	77.62	52.29↓	6.49	2.96↓	34.5	15.83↓
Claude-haiku-03	67.59	40.68↓	67.78	44.35↓	3.48	0.45↓	22.82	4.19↓
Gemini 1.5 Pro	76.6	54.72↓	75.63	50.67↓	6.84	1.54↓	34.52	13.6↓
Gemini Flash	69.46	53.12↓	70.76	51.18↓	5.61	1.35↓	29.7	11.86↓
Gemini Flash-8B	70.51	49.99↓	71.07	51.42↓	4.1	0.78↓	27.63	6.33↓
Llama3.1-8B	55.32	44.37↓	62.88	46.72↓	0.63	0.06↓	14.44	5.39↓
Gemma2-9B	64.36	47.94↓	68.18	50.22↓	1.83	0.51↓	20.08	8.18↓

mantic mismatches, producing grammatically correct sentences that are unrelated to the original input content or topic.

MSA to ALG Without linguistic features, several recurrent error types are observed. First, outputs sometimes retain MSA vocabulary instead of generating dialectal equivalents, reflecting incomplete dialect adaptation. Second, models occasionally produce lexical words from other Arabic dialects, most commonly Moroccan dialect.

Orthographic distortions are also frequent, where minor character alterations lead to incorrect word forms (e.g., generating *تعام* (*food*) instead of *طعام* or *خطارت* (*choose*) instead of *اخترت*). Additionally, models sometimes reorder internal characters or syllables, resulting in awkward or nonstandard pronunciation patterns.

Another common issue involves grammatical

agreement, particularly gender mismatches, where model incorrectly switch between masculine and feminine forms during translation.

When POS tags are introduced, a distinct class of semantic errors emerges. Models sometimes confuse syntactic roles, especially between subjects and objects, leading to outputs that are linguistically fluent and dialectally appropriate but semantically incorrect. This indicates that while linguistic features may improve surface quality, they can also disrupt deeper sentence-level interpretation.

6. Conclusion

This study used LLMs to investigate how linguistic augmentation affects bidirectional translation between Algerian dialect and MSA. Findings indicate that auxiliary features like POS tags and diacritics frequently cause noise instead of enhancing performance, with asymmetric effects in both directions:

Table 7: Human evaluation for MSA→ALG (Adequacy and Fluency). Arrows indicate change relative to w/o.

Model	Adequacy			Fluency		
	w/o	w/ POS	w/ DIACT	w/o	w/ POS	w/ DIACT
GPT-4 models*	88.63	34.74↓	92.95↑	93.05	31.26↓	87.26↓
Claude-3-5-sonnet-10	93.68	67.05↓	88.74↓	88.74	58.84↓	81.37↓
Google-gemini-1.5-pro	91.46	45.26↓	94.53↑	87.58	37.05↓	87.26↓
Gemma2:9b-instruct-q5	4.74	3.37↓	4.00↓	4.74	2.32↓	2.21↓
Llama3.1:8b-instruct-q5	1.58	1.47↓	2.84↑	1.58	0.42↓	3.05↑

Table 8: Human evaluation for ALG→MSA (Adequacy and Fluency). Arrows indicate change relative to w/o.

Model	Adequacy			Fluency		
	w/o	w/ POS	w/ DIACT	w/o	w/ POS	w/ DIACT
GPT-4 models*	82.70	65.05↓	45.26↓	95.05	56.42↓	38.00↓
Claude-3-5-sonnet-10	80.42	67.58↓	72.53↓	96.42	59.26↓	65.89↓
Google-gemini-1.5-pro	52.32	61.68↑	53.89↑	90.74	57.68↓	43.37↓
Gemma2:9b-instruct-q5	6.84	6.00↓	5.26↓	6.95	4.42↓	4.42↓
Llama3.1:8b-instruct-q5	1.89	0.21↓	0.53↓	1.89	0↓	0.32↓

diacritics can improve adequacy in MSA→ALG but significantly hinder translation in ALG→MSA because of distribution mismatch. Additionally, we see distinct differences in capability between models: smaller open-weight models exhibit significant performance degradation, while frontier LLMs show strong dialect comprehension but are still limited in dialect generation. Lastly, a significant discrepancy between automated metrics and human assessment is revealed, with contextual semantic metrics outperforming traditional overlap-based measures in capturing adequacy changes.

7. References

- Ahmed Elmogtaba Abdelmoniem Ali Abdelaziz, Ashraf Hatim Elneima, and Kareem Darwish. 2024. Llm-based mt data creation: Dialectal to msa translation shared task. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation@ LREC-COLING 2024*, pages 112–116.
- Muhammad Abdul-Mageed and Houda Bouamor. 2021. Arabic text processing with morphological and diacritical features. *Computational Linguistics*.
- Muhammad Abdul-Mageed, AbdelRahim Elmadany, et al. 2021. Arbert & marbert: Deep bidirectional transformers for arabic. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7088–7105.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florenzia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Abdullah Alabdullah, Lifeng Han, and Chenghua Lin. 2025. Advancing dialectal arabic to modern standard arabic machine translation. *arXiv preprint arXiv:2507.20301*.
- Sawsan Alqahtani, Mahmoud Ghoneim, and Mona Diab. 2016. Investigating the impact of various partial diacritization schemes on arabic-english statistical machine translation. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers’ Track*, pages 191–204.
- AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1.
- Hanin Atwany, Nour Rabih, Ibrahim Mohammed, Abdul Waheed, and Bhiksha Raj. 2024. Osact 2024 task 2: Arabic dialect to msa translation. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation@ LREC-COLING 2024*, pages 98–103.
- Baligh Babaali, Mohammed Salem, and Nawaf R Alharbe. 2025. Breaking language barriers with

- chatgpt: enhancing low-resource machine translation between algerian arabic and msa. *International Journal of Information Technology*, 17(7):4109–4118.
- Laith H Baniata, Seyoung Park, and Seong-Bae Park. 2018. A multitask-based neural machine translation model with part-of-speech tags integration for arabic dialects. *Applied Sciences*, 8(12):2502.
- Reem Bassiouney. 2009. *Arabic Sociolinguistics*. Edinburgh University Press.
- Mohamed Benrabah. 2014. [Competition between four "world" languages in algeria](#). *Journal of World Languages*, 1(1):38–59.
- Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, et al. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.
- Farida Bouhadiba. 2017. Algerian arabic: A sociolinguistic overview. In Elabbas Benmamoun and Reem Bassiouney, editors, *The Routledge Handbook of Arabic Linguistics*, pages 353–368. Routledge.
- Mohamed Amine Cheragui, Abdelhalim Hafedh Dahou, and Amin Abdedaïem. 2023. Exploring bert models for part-of-speech tagging in the algerian dialect: A comprehensive study. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 140–150.
- Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Anna Currey and Kenneth Heafield. 2020. Incorporating linguistic annotations into neural machine translation. In *Proceedings of WMT*.
- Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1070–1074.
- Mona Diab, Mahmoud Ghoneim, and Nizar Habash. 2007. Arabic diacritization in the context of statistical machine translation. In *Proceedings of Machine Translation Summit XI: Papers*.
- Mona Diab and Kadri Hacioglu. 2004. Automatic diacritization of arabic text. In *Proceedings of COLING*.
- Lukas Edman and Anders Sogaard. 2023. Do large language models benefit from linguistic supervision? In *Proceedings of ACL*.
- Ashraf Hatim Elneima, AhmedElmogtaba Abdelmoniem Ali Abdelaziz, and Kareem Darwish. 2024. Osact6 dialect to msa translation shared task overview. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation@ LREC-COLING 2024*, pages 93–97.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool.
- Nizar Habash. 2013. *Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies.
- Nizar Habash. 2019. *Arabic Natural Language Processing*. Cambridge University Press.
- Nizar Habash and Owen Rambow. 2007. Arabic diacritization through full morphological tagging. In *Proceedings of NAACL*.
- Clive Holes. 2004. *Modern Arabic: Structures, Functions, and Varieties*, revised edition. Georgetown University Press.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. *arXiv preprint arXiv:2103.06678*.
- Salam Khalifa et al. 2022. Camel tools: An open platform for arabic natural language processing. In *Proceedings of LREC*.
- Abdullah Salem Khered, Youcef Benkhedda, and Riza Theresa Batista-Navarro. 2025. Dial2msa-verified: A multi-dialect arabic social media dataset for neural machine translation to modern standard arabic. In *Proceedings of the 4th Workshop on Arabic Corpus Linguistics (WACL-4)*, pages 50–62.
- Djamel Eddine Kouloughli. 1999. *Le maghrébin*. Presses Universitaires de France.

- Yinhan Liu and Graham Neubig. 2024. Revisiting explicit syntax in neural machine translation. *Transactions of the ACL*.
- Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. 2015. Machine translation experiments on padic: A parallel arabic dialect corpus. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, pages 26–34.
- Rasha Obeidat, Luay Alawneh, and Yara Al-Harabsheh. 2025. Exploring prompting for dialectal machine translation: a focus on north jordanian arabic. *PeerJ Computer Science*, 11:e3209.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of LREC*.
- Lotfi Sayahi. 2014. *Diglossia and Language Contact: Language Variation and Change in North Africa*. Cambridge University Press.
- Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91.
- Serry Sibae, Omer Nacar, Yasser Al-Habashi, Adel Ammar, and Wadii Boulila. 2025. Shami-mt: a syrian arabic dialect to modern standard arabic bidirectional machine translation system. *arXiv preprint arXiv:2508.02268*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Kees Versteegh. 2014. *The Arabic Language*, 2 edition. Edinburgh University Press.
- Nasser Zalmout and Nizar Habash. 2020. Adversarial neural arabic diacritization. In *Proceedings of ACL*.
- Imed Zitouni, Jeffrey Sorensen, and Ruhi Sarikaya. 2006. Arabic diacritization using a statistical approach. In *Proceedings of HLT-NAACL*.

ASCAT: Arabic Scientific Benchmark for Advanced Translation Evaluation

Serry Sibae¹, Khlood Al Jallad², Zineb Yousfi³, Israa Elhosiny³,
Yousra El-Ghawi³, Batool Balah³, Omer Nacar⁴

¹Prince Sultan University, ²SySSR

³NAMAA Community, ⁴Tuwaiq Academy

ssibae@psu.edu.sa, k.jallad.1@gmail.com, yousfi.zineb.yz@gmail.com,
israaelhosiny@gmail.com, yousraghawi@gmail.com, batoolnajeh@gmail.com,
o.nacar@tuwaiq.edu.sa

Abstract

We present ASCAT (Arabic Scientific Corpus for Advanced Translation), a high-quality English-Arabic parallel benchmark corpus designed for scientific translation evaluation constructed through a systematic multi-engine translation and expert post-editing pipeline. Unlike existing Arabic-English corpora that rely on short sentences or single-domain text, ASCAT targets full scientific abstracts averaging 125.3 words (English) and 111.78 words (Arabic), drawn from five scientific domains: physics, mathematics, computer science, quantum mechanics, and artificial intelligence. Each abstract was translated using three complementary architectures: generative AI (Gemini), transformer-based models (Hugging Face `quickmt-en-ar`), and commercial MT API (Google Translate), and subsequently post-edited by domain experts at the lexical, syntactic, and semantic levels. The resulting corpus contains 67,293 English tokens and 60,026 Arabic tokens, with an Arabic vocabulary of 17,604 unique words reflecting the morphological richness of the language. We benchmark three state-of-the-art LLMs on the corpus: GPT-4o-mini (BLEU: 37.07), Gemini-3.0-Flash-Preview (BLEU: 30.44), and Qwen3-235B-A22B (BLEU: 23.68) demonstrating its discriminative power as an evaluation benchmark. ASCAT addresses a critical gap in scientific MT resources for Arabic and is designed to support rigorous evaluation of scientific translation quality and training of domain-specific translation models.

Keywords: Arabic machine translation, parallel corpus, scientific translation, multi-engine translation, neural machine translation, English-Arabic NLP, domain-specific corpus, corpus construction

1. Introduction

The rapid growth of scientific literature has intensified the need for reliable domain-specific translation models, particularly for Arabic spoken by over 400 million people yet grossly underrepresented in scientific discourse. This language gap creates a critical accessibility barrier for Arabic-speaking researchers and professionals. A key bottleneck in improving Arabic scientific machine translation (MT) is the scarcity of high-quality parallel corpora that maintain terminological accuracy and conceptual consistency.

This paper presents the construction and analysis of an English-Arabic scientific translation corpus built through a systematic multi-engine machine translation and expert post-editing process. By combining multiple state-of-the-art translation engines including generative AI (Gemini), transformer-based models (Hugging Face), and commercial MT APIs (Google Translate, DeepL) with rigorous expert post-editing, our approach balances scalability with quality. The corpus covers full scientific abstracts across diverse domains including physics, mathematics, computer science, quantum mechanics, and artificial intelligence, averaging 125.3 words (English) and 111.8 words (Arabic) per abstract far exceeding the complexity of existing

datasets. Critically, ASCAT is explicitly designed as an evaluation benchmark rather than a large-scale training dataset, prioritizing depth of validation over dataset size.

2. Background

Existing English-Arabic parallel corpora suffer from several key limitations. General-domain resources such as MultiUN (Eisele and Chen, 2010) and OPUS (Tiedemann, 2012) lack the terminological precision required for scientific translation. Domain-specific datasets like DEAST (Author and Others, 2026) (33,000 thesis title pairs, ~9 words average) and PEACH (Al-Sabbagh, 2024) (51,671 healthcare sentence pairs, ~10–12 words average) are too short to capture the syntactic complexity and discourse-level phenomena of scientific abstracts. While Tarjama-25 (Hennara et al., 2025) targets longer sentences (~75 words), its 5,000-sentence scale is insufficient for training large MT models. ATHAR (Mohammed and Khalil, 2025) addresses classical Arabic scientific texts, which differ substantially in register from modern scientific writing. Table 1 provides a structured comparison of these corpora against our dataset.

Multi-engine translation approaches have emerged as a promising strategy, exploiting

Table 1: Comparison of English–Arabic Parallel Corpora

Dataset	Domain	Size	Type	Avg. Len.	Val.
DEAST (Author and Others, 2026)	Multi-sci.	33k	Titles	9 w	Expert
PEACH (Al-Sabbagh, 2024)	Health	51.7k	Leaflets	10–12 w	Experts
ATHAR (Mohammed and Khalil, 2025)	Classical	66k	Historical	–	Expert
Tarjama-25 (Hennara et al., 2025)	Multi-dom.	5k	Long sent.	75 w	Prof.
ASCAT (Ours)	Multi-sci.	500	Abstracts	125 / 112 w	Multi-stage

complementary strengths: statistical MT excels at terminology consistency, neural MT produces fluent output, and large language models handle contextual and idiomatic expressions. Combining their outputs enables comparative analysis and focuses human validation on high-risk segments. Human validation remains essential in scientific translation: full post-editing, selective validation guided by confidence scoring, and inter-annotator agreement metrics each play a role in ensuring quality, yet few existing corpora document their validation protocols transparently.

Our dataset addresses these gaps by targeting full scientific abstracts, employing a multi-engine translation pipeline, and applying multi-stage expert post-editing yielding a resource suitable for both training domain-specific MT models and benchmarking scientific translation quality.

3. Methodology

The dataset construction follows a three-stage pipeline: Data Collection, Multi-Engine Translation, and Expert Post-editing as illustrated in Figure 1.

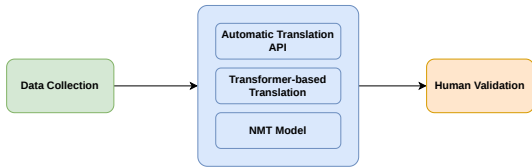


Figure 1: Dataset construction pipeline: Data Collection, Multi-Engine Translation, and Human Validation.

3.1. Data Collection

Scientific abstracts were systematically gathered across multiple domains, including physics, mathematics, computer science, quantum mechanics, and artificial intelligence. To ensure representational diversity and avoid selection bias, abstracts were randomly sampled from papers across these domains, yielding a balanced multi-domain corpus of full-length scientific abstracts.

The abstracts were sourced from arXiv, randomly selected from the gfiisore/arxiv-abstracts-2021 dataset, which contains scientific papers from

arXiv. arXiv is an open-access repository that allows redistribution of abstracts under its licensing terms, as the content is openly available for research and educational purposes.

3.2. Preprocessing

To ensure consistency and handle scientific content, we applied minimal preprocessing to the English abstracts before translation. LaTeX expressions and mathematical symbols were preserved in their original English form, as they are universally understood in scientific contexts and do not require translation. Scientific symbols (e.g., Greek letters, operators) were similarly kept unchanged. No automated normalization was applied to the text after translation; any adjustments to LaTeX or symbols were made individually by reviewers during the post-editing phase if deemed necessary for clarity or accuracy.

The preprocessing pipeline consisted of:

1. Extraction of plain text from the arXiv abstracts.
2. Random selection of samples for translation.
3. Preservation of LaTeX and scientific symbols.

3.3. Multi-Engine Translation

To maximize linguistic diversity and enable comparative analysis, each abstract was translated using three distinct architectures in the following order: first Gemini, then Hugging Face `quickmt-en-ar` and Google Translate. Samples were randomly selected for translation without additional filtering beyond domain balance.

3.4. Human Validation Process

The final stage involved comprehensive expert review and post-editing of all translated samples. Post-editing was conducted by seven domain experts, each holding at minimum an undergraduate-level degree in either Arabic linguistics or a relevant scientific discipline (physics, mathematics, computer science, or AI).

Validators worked independently using a structured checklist, spending an average of 10-30 minutes per sample. The post-editing process involved

four main types of edits: substitutions (replacing incorrect terms), insertions (adding missing information), deletions (removing redundant or erroneous content), and rewrites (restructuring sentences for better fluency and accuracy).

Since we distributed the abstracts to reviewers and did not conduct formal inter-annotator agreement (IAA), our validation focused on qualifying the quality of the translations and fixing errors rather than measuring inter-annotator consistency. This decision was intentional, as our primary goal was to produce high-quality references for benchmarking (see Table 2).

Table 2: Human Validation Checklist

Level	Criterion	Binary (Y/N)
Lexical	Domain terminology	Y
Lexical	Preservation of NEs	Y
Syntactic	Grammatical correctness	Y
Syntactic	Sentence structure	Y
Semantic	Epistemic hedging	N
Discourse	Sentence consistency	N

Expert post-editors with expertise in both Arabic linguistics and the relevant scientific domains post-edited translations at the lexical, syntactic, and semantic levels, correcting terminology errors, structural inconsistencies, and meaning deviations. Concrete examples of edits include:

- **Substitution:** Changing "machine learning" to (al-taallum al-ālī) for accurate domain terminology.
- **Insertion:** Adding (fī majāl) to specify "in the field of" for clarity.
- **Deletion:** Removing redundant particles that do not contribute to meaning.
- **Rewrite:** Restructuring awkward sentences to improve flow, e.g., converting passive voice to active where appropriate in Arabic.

A detailed analysis of the most frequent translation errors identified during this phase is provided in Section 4.

4. Corpus Statistics

4.1. Domain Coverage

The dataset was strategically curated to ensure broad coverage of modern scientific disciplines. The corpus spans five primary domains: Quantum Mechanics, Artificial Intelligence, Computer Science, Mathematics, Physics. This multi-domain design ensures that the corpus captures the terminological diversity and stylistic variation inherent to different scientific fields, making it suitable for

training and evaluating general-purpose scientific MT systems rather than narrow single-domain models. The deliberate balance across domains also mitigates the risk of domain-specific lexical bias, where a model trained predominantly on one field may underperform on others.

4.2. Abstract Length Analysis

To assess the linguistic complexity of the corpus, we analyzed word and character distributions across all abstract pairs. Table 3 summarizes the key statistics for both the English source and Arabic target abstracts.

Table 3: Abstract Length Statistics

Lang	Min	Med	Mean	Max	SD	Mean Ch.
EN	3	113	125.3	297	64.0	822
AR	4	100	111.8	315	58.9	695

The English abstracts average 125.31 words per abstract (median: 113), with a standard deviation of 63.99 words, indicating substantial length variability across domains and paper types. Arabic translations average 111.78 words (median: 100) with a standard deviation of 58.87. The consistently lower word count in Arabic relative to English is linguistically expected, as Arabic's rich morphological system and agglutinative properties allow it to encode more information per word through affixation and cliticization.

The high standard deviation in both languages reflects the natural heterogeneity of scientific abstracts: concise theoretical results may be presented in fewer than 50 words, while comprehensive experimental studies may span close to 300 words. This variability is a desirable property for a training corpus, as it exposes MT models to a wide spectrum of abstract lengths and structural complexities.

4.3. Vocabulary and Lexical Richness

Table 4 presents the vocabulary and token-level statistics for both language sides of the corpus.

Table 4: Vocabulary Statistics

Lang	Tokens	Unique words	TTR	Mean Sent.
EN	67,293	12,685	0.19	7.16
AR	60,026	17,604	0.29	6.99

Several notable observations emerge from the vocabulary analysis. First, despite having fewer total tokens (60,026 vs. 67,293), the Arabic side exhibits a substantially larger vocabulary size (17,604 unique words vs. 12,685 in English). This is reflected in the Type-Token Ratio (TTR) (Richards and Schmidt, 2013), a standard measure of lexical

diversity, where Arabic scores 0.2933 compared to English’s 0.1885. The higher Arabic TTR is consistent with the morphological complexity of Arabic, where a single root can generate dozens of surface forms through derivational and inflectional processes, resulting in a much larger effective vocabulary space. This has direct implications for MT model design, as Arabic-side models require larger vocabularies or subword tokenization strategies (e.g., BPE or SentencePiece) to adequately cover the target language’s lexical space.

Second, both language sides exhibit comparable mean sentence counts per abstract (7.16 for English and 6.99 for Arabic), confirming that the translation process preserved the discourse segmentation of the original abstracts without undue merging or splitting of sentences. This structural fidelity is important for training models that must learn to produce coherent, multi-sentence scientific discourse rather than isolated sentence translations.

5. Evaluation

The following evaluation demonstrates ASCAT’s utility as a discriminative benchmark, a role for which validation quality and domain complexity matter more than corpus size.

To assess the quality of our corpus and benchmark the performance of state-of-the-art translation systems on scientific English-Arabic translation, we evaluated three large language models against our human-validated reference translations. The models evaluated are **GPT-4o-mini** (OpenAI), **Gemini-3.0-Flash-Preview** (Google DeepMind), and **Qwen3-235B-A22B** (Alibaba Cloud). Each model was prompted to translate the English source abstracts, and the outputs were scored against the human-validated Arabic references using two standard automatic evaluation metrics: BLEU and ROUGE.

5.1. Results

Table 5 presents the automatic evaluation results for all three models.

Table 5: Automatic Evaluation on EN–AR Scientific Translation

Model	BLEU ↑	R-1 ↑	R-2 ↑	R-L ↑
GPT-4o-mini	37.07	0.590	0.476	0.586
Gem-3.0-Flash	30.44	0.530	0.390	0.522
Qwen3-235B	23.68	0.537	0.410	0.531

5.2. Discussion

GPT-4o-mini achieves the highest BLEU (37.07), indicating strong alignment with expert-validated

references (Yan et al., 2024). BLEU scores above 30 signify high-quality translations in Arabic, reflecting robust n-gram overlap in scientific contexts.

Gemini-3.0-Flash-Preview (BLEU: 30.44) shows adequate content coverage but less precise matching, while Qwen3-235B (BLEU: 23.68) performs competitively on ROUGE-1 yet structurally distant.

The 13.4 BLEU gap demonstrates the corpus’s discriminative power. Compared to DEAST’s 2.15–3.36 BLEU on short titles (Author and Others, 2026), our 23.68–37.07 scores highlight superior quality for benchmarking advanced MT on long-form text.

Error analysis revealed issues in terminology, structure, and semantics, emphasizing scientific MT challenges and expert post-editing value.

Moderate scores across models underscore the difficulty of scientific Arabic translation, justifying domain-specific resources like ASCAT.

6. Limitations and Future Work

Limitations include the dataset’s size (500 abstracts, prioritized for quality), uneven domain distribution, and reliance on automatic metrics that miss qualitative aspects like semantic nuance.

Scientific translation challenges include terminological ambiguity, non-standardized terms, epistemic hedging, and acronyms without Arabic equivalents.

Future work: expand corpus for balanced domains, add human evaluation, fine-tune models for improved translation, conduct per-domain evaluation, log edit-level statistics, and analyze lemmatized TTR.

7. Conclusion

We presented ASCAT, a high-quality English-Arabic parallel benchmark corpus for scientific translation, constructed via multi-engine MT and expert post-editing. It addresses gaps in Arabic scientific MT with abstract-level complexity, multi-domain coverage, and transparency.

Analysis shows Arabic’s morphological richness (TTR 0.29, vocab 17,604). Benchmarking revealed a 13.4 BLEU gap, proving discriminative power while highlighting translation challenges.

ASCAT advances Arabic scientific communication and supports domain-specific MT research.

8. Acknowledgments

We would like to thank the NAMAA community for their support in this project.

9. Bibliographical References

- Rania Al-Sabbagh. 2024. [Peach: A sentence-aligned parallel english-arabic corpus for health-care](#). *Corpora*, 19(3):395–410.
- M. A. Author and Others. 2026. [Deast: A dataset for english-arabic scientific translation and vice versa](#). *Data in Brief*, 64:112381.
- Andreas Eisele and Yu Chen. 2010. [MultiUN: A multilingual corpus from united nation documents](#). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).
- K. Hennara, M. Hreden, M. M. Hamed, Z. Aldallal, S. Chrouf, and S. AlModhayan. 2025. Mutarjim: Advancing bidirectional arabic-english translation with a small language model. *arXiv preprint arXiv:2505.17894*.
- Mahmoud S. Mohammed and Mohamed Khalil. 2025. Athar: A high-quality and diverse dataset for classical arabic to english translation. In *Proceedings of the Third Arabic Natural Language Processing Conference*, pages 97–106.
- Jack C Richards and Richard W Schmidt. 2013. *Longman dictionary of language teaching and applied linguistics*. Routledge.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of LREC 2012*, pages 2214–2218.
- Jianhao Yan, Pingchuan Yan, Yulong Chen, Judy Li, Xianchao Zhu, and Yue Zhang. 2024. [Gpt-4 vs. human translators: A comprehensive evaluation of translation quality across languages, domains, and expertise levels](#).

SHEINfer: Implicit Product Category Inference from Arabic E-commerce Reviews

Hend Al-Khalifa

College of Computer and Information Sciences, King Saud University
Riyadh, Saudi Arabia
hendk@ksu.edu.sa

Abstract

We introduce SHEINfer, a novel task and dataset for inferring product categories from Arabic e-commerce reviews without explicit product mentions. Unlike traditional product classification that relies on product titles or descriptions, our task requires models to deduce product types solely from customer review text, which often contains implicit references through dialectal expressions, quality assessments, and contextual clues. We present a dataset of 801 Arabic reviews from the SHEIN e-commerce website, dual-annotated across 11 product categories with 515 agreed samples achieving moderate inter-annotator agreement (Cohen's $\kappa = 0.60$). Given the relatively small dataset size, we employ 5-fold stratified cross-validation for all models to ensure robust performance estimates. Our experiments compare traditional machine learning approaches (TF-IDF with SVM and Logistic Regression), Arabic transformer models (AraBERT, CAMELBERT, MARBERT), and large language models (GPT-4o-mini) in zero-shot and few-shot settings. Results show that MARBERT achieves the highest accuracy (0.586 ± 0.026), while TF-IDF with Logistic Regression achieves the best macro F1 (0.417 ± 0.056), indicating better performance across minority categories. GPT-4o-mini demonstrates poor zero-shot performance (0.064 accuracy) with modest improvement in 3-shot settings (0.186 accuracy), indicating that implicit product inference from dialectal Arabic text remains challenging for general-purpose LLMs. Our findings highlight the unique challenges of implicit product classification in Arabic e-commerce and establish benchmarks for future research in this underexplored area.

Keywords: Product Classification, E-commerce, Implicit Inference, Dialectal Arabic, Arabic NLP, Large Language Models

1. Introduction

The rapid growth of e-commerce platforms in the Middle East and North Africa (MENA) region has generated vast amounts of Arabic user-generated content. Customer reviews on platforms like SHEIN¹, Amazon², and Noon³ provide valuable insights into consumer preferences and product characteristics. However, these reviews present unique challenges for natural language processing due to the prevalence of dialectal Arabic, informal writing styles, and implicit product references.

Traditional product classification tasks assume access to explicit product information such as titles, descriptions, or metadata. In contrast, real-world scenarios often require inferring product categories from customer reviews alone, where products are referenced implicitly through quality descriptions, fit assessments, or contextual clues. For example, a review stating "الخلعة باردة وصيفية" (the fabric is cool and summery) implicitly refers to clothing without explicit mention.

We introduce SHEINfer, a task and dataset specifically designed to address implicit product category inference from Arabic e-commerce reviews. Our contributions include: (1) A novel task formulation that challenges models to infer product categories without explicit product mentions; (2) A carefully curated dataset of 801 Arabic reviews with dual annotation across 11

product categories; (3) Comprehensive experiments comparing traditional ML, Arabic transformers, and large language models including GPT-4o-mini; (4) Analysis revealing that implicit product inference remains challenging even for state-of-the-art LLMs, with traditional ML approaches achieving competitive performance on macro F1.

The rest of the paper is organized as follows: Section 2 reviews related work on Arabic NLP, product attribute extraction, and large language models for e-commerce. Section 3 formally defines the implicit product category inference task. Section 4 describes our dataset, including the data source, annotation process, product categories, and inter-annotator agreement. Section 5 details our methodology, covering text preprocessing, the models evaluated, and the experimental setup. Section 6 presents the experimental results with performance comparisons across all models, it also provides a detailed discussion of the results, including analysis of traditional ML performance, transformer models, LLM performance, and a comprehensive error analysis. Section 7 concludes the paper with a summary of findings and directions for future work and acknowledges the limitations of our study.

¹ <https://shein.com>

² <https://www.amazon.com/>

³ <https://www.noon.com>

2. Related Work

Arabic NLP has witnessed significant advances with the development of pre-trained language models. AraBERT (Antoun et al., 2020) introduced BERT-based pre-training for Arabic using news articles and Wikipedia, while CAMeLBERT (Inoue et al., 2021) specifically addressed dialectal Arabic through pre-training on diverse Arabic varieties including Gulf, Levantine, and Egyptian dialects. MARBERT (Abdul-Mageed et al., 2021) further advanced dialectal Arabic understanding by pre-training exclusively on Arabic Twitter data, capturing informal language patterns prevalent in social media and e-commerce contexts. These models have achieved state-of-the-art results on various Arabic NLP benchmarks including sentiment analysis, named entity recognition, and question answering. Arabic sentiment analysis in e-commerce has been extensively studied, with Al-Smadi et al. (2018) developing aspect-based sentiment analysis for Arabic hotel reviews using deep recurrent neural networks. Elnagar et al. (2018) created large-scale Arabic book review datasets (LABR) with over 63,000 reviews spanning multiple sentiment categories. More recently, ensemble approaches combining AraBERT with deep learning models have achieved over 92% accuracy on Arabic e-commerce sentiment classification (Habbat et al., 2023). However, these works focus on sentiment polarity rather than product category inference.

Product attribute extraction has emerged as a critical task in e-commerce NLP. OpenTag (Zheng et al., 2018) pioneered deep learning approaches for open attribute value extraction from product profiles using BiLSTM-CRF architectures with attention mechanisms, achieving 83% F1-score while discovering new attribute values from minimal annotations. Wang et al. (2020) reformulated attribute extraction as a question-answering task using multi-task learning, improving generalizability across attributes. More recently, PAM (Lin et al., 2021) introduced multimodal approaches combining product images with text for cross-category attribute extraction, demonstrating 15% improvement over text-only methods. The MAVE dataset (Yang et al., 2022) provided large-scale multi-source attribute value annotations enabling systematic benchmarking of extraction methods.

Large language models have revolutionized product attribute extraction. Blume et al. (2023) demonstrated that generative models can extract both explicit and implicit attributes from product descriptions, outperforming sequence tagging methods while achieving greater data efficiency. Their work on Amazon and MAVE datasets showed LLMs' unique capability for implicit attribute extraction that traditional NER models cannot perform. Fang et al. (2024) proposed LLM-Ensemble, combining multiple LLMs for optimal

attribute value extraction at Walmart, achieving state-of-the-art performance through ensemble methods. Zou et al. (2024a) introduced ImplicitAVE, the first multimodal dataset specifically designed for implicit attribute value extraction, benchmarking various multimodal LLMs including GPT-4V, BLIP-2, and LLaVA. Their companion work EIVEN (Zou et al., 2024b) presented an efficient multimodal LLM framework for implicit attribute extraction with reduced reliance on labeled data.

Product classification typically relies on product titles, descriptions, or images. Zahera and Sherif (2020) proposed ProBERT, a multi-label BERT architecture for product categorization achieving over 90% accuracy on e-commerce datasets. Recent work by Çiftlikçi et al. (2025) compared LLMs and deep learning models for Turkish e-commerce attribute extraction, demonstrating LLMs' superior performance in handling complex linguistic structures. Zhou et al. (2024) introduced decorative relation correction with LLAMA 2.0-based annotation for enhanced e-commerce attribute recognition. Our work differs fundamentally from prior research by focusing on inferring product categories from customer reviews without access to product metadata. This implicit inference task requires understanding contextual clues, domain knowledge, and dialectal expressions that characterize Arabic e-commerce discourse, representing a novel challenge distinct from both explicit attribute extraction and traditional product classification.

3. Task Definition

Given a customer review text R in Arabic (potentially dialectal), the task is to predict the product category C from a predefined set of 11 categories. The key challenge is that reviews typically do not explicitly mention product names or categories. Instead, models must infer the product type from implicit cues such as: (1) Quality descriptors ("القماش ناعم" / the fabric is soft → clothing); (2) Fit and sizing references ("المقاس مناسب" / the size fits → clothing/shoes); (3) Functional descriptions ("الشحن سريع" / charging is fast → electronics); (4) Aesthetic assessments ("اللون حلو" / the color is nice → various categories with contextual disambiguation).

This task formulation reflects real-world scenarios where product metadata may be unavailable, corrupted, or when analyzing reviews aggregated across platforms. It also tests models' abilities to understand Arabic dialectal expressions and perform implicit reasoning about product characteristics.

4. Dataset

4.1 Data Source

We utilize the Arabic Reviews of SHEIN dataset (Bin Safi, 2024) available on Hugging Face, containing 2,415 Arabic product reviews from the SHEIN e-commerce platform. The original dataset includes raw review text, cleaned text with emojis and repeated characters removed, and star ratings (1-5). Reviews predominantly feature Saudi Arabian dialect with influences from other Gulf and Levantine dialects, reflecting SHEIN's diverse Arabic-speaking customer base.

4.2 Annotation Process

Two annotators independently labelled 801 reviews for product categories on a voluntary basis. Both annotators hold bachelor's degrees in information technology and are native Arabic speakers with extensive familiarity with the SHEIN platform as regular customers. This domain expertise was crucial for understanding the contextual clues and dialectal expressions commonly used in Arabic e-commerce reviews. Annotators received detailed guidelines with category definitions and examples of implicit references for each category. A calibration session was conducted on pilot samples to ensure consistent understanding and resolve ambiguous cases. Each annotator then worked independently to prevent bias, assigning exactly one category per review based on contextual inference from the review text alone, without access to product images or metadata.

4.3 Product Categories

We defined 11 product categories based on SHEIN's catalog structure: Makeup/Beauty (مكياج/تجميل), Accessories (اكسسوارات), Unclear (غير واضح), Shoes (حذاء), Dress/Abaya (فسطان/عباية), Home Products (منتجات منزلية), Pants/Jean (بنطلون/جينز), Blouse/Shirt (بلوزة/قميص), Underwear/Sportswear (ملابس داخلية/رياضية), Skirt (تنورة), and Children's Clothing (ملابس أطفال). The "Unclear" category captures reviews where even human annotators could not determine the product type due to insufficient contextual information. Table 1 shows dataset statistics by category.

Category	Count	Percentage	Avg. Length (words)
Makeup/Beauty (مكياج/تجميل)	133	25.8%	11.8
Accessories (اكسسوارات)	68	13.2%	12.6
Unclear (غير واضح)	64	12.4%	5.8
Shoes (حذاء)	63	12.2%	15.7
Dress/Abaya (فسطان/عباية)	50	9.7%	15.1

Category	Count	Percentage	Avg. Length (words)
Home Products (منتجات منزلية)	44	8.5%	11.1
Pants/Jean (بنطلون/جينز)	28	5.4%	17.4
Blouse/Shirt (بلوزة/قميص)	26	5.0%	10.4
Underwear/Sportswear (ملابس داخلية/رياضية)	23	4.5%	16.0
Skirt (تنورة)	11	2.1%	12.5
Children's Clothing (ملابس أطفال)	5	1.0%	7.2
Total	515	100%	12.3

Table 1: Dataset Statistics by Category

4.4 Inter-Annotator Agreement

We computed Cohen's Kappa (κ) to measure inter-annotator agreement, achieving $\kappa = 0.60$, indicating moderate agreement (Landis & Koch, 1977). Of 801 annotated reviews, 515 (64.3%) achieved full agreement between annotators and form our experimental dataset. The moderate kappa score reflects the inherent difficulty of the task, as many reviews contain ambiguous or minimal contextual clues. Disagreements primarily occurred in cases where reviews could plausibly refer to multiple categories (e.g., "حلو ومريح" / nice and comfortable could apply to clothing or shoes) or when reviews contained only generic praise without product-specific descriptors.

5. Methodology

5.1 Text Preprocessing

We apply standard Arabic text preprocessing including: removal of URLs and web links, normalization of Arabic characters (أ → آ, إ → إ, ي → ي, ه → ه), and whitespace normalization to collapse multiple spaces into single spaces. For transformer models, we use the tokenizers provided by each model's respective library without additional preprocessing, allowing the pre-trained models to handle text in their expected format.

5.2 Models

- Traditional Machine Learning:** We implement TF-IDF vectorization with unigrams and bigrams, maximum 5000 features, minimum document frequency of 2, and maximum document frequency of 0.95. We combine this with Support Vector Machine (SVM) using RBF kernel with balanced class weights, and Logistic

Regression with multinomial loss, balanced class weights, and L-BFGS solver.

2. **Arabic Transformer Models:** We fine-tune three Arabic BERT variants: (1) AraBERT-v2 (aubmindlab/bert-base-arabertv2), pre-trained on Modern Standard Arabic (MSA) from news articles; (2) CAMEL-BERT-mix (CAMEL-Lab/bert-base-arabic-camelbert-mix), pre-trained on a mixture of MSA and dialectal Arabic including Gulf dialects; and (3) MARBERT (UBC-NLP/MARBERT), pre-trained exclusively on 1 billion Arabic tweets, making it particularly suited for informal and dialectal Arabic text common in e-commerce reviews. All models are trained for 3 epochs with batch size 16, learning rate $3e-5$, 50 warmup steps, and weight decay of 0.01.
3. **Large Language Model (GPT-4o-mini):** Following Brown et al. (2020), we evaluate in-context learning capabilities of GPT-4o-mini via the OpenAI API with rate limiting. Zero-shot prompts provide task description and category list without examples. Few-shot (3-shot) prompts include three labeled examples per category sampled from the training fold. We use temperature=0 for deterministic outputs.

5.3 Experimental Setup

Given the relatively small size of our dataset (515 samples), we employ 5-fold stratified cross-validation for all models, including GPT-4o-mini, to ensure robust and comparable performance estimates. This choice is motivated by several considerations. First, cross-validation provides more robust and reliable performance estimates by using all available data for both training and testing across different folds, reducing the variance that would result from a single random split. Second, with only 515 samples distributed across 11 imbalanced categories, a single train/test split could result in some minority categories having very few or no samples in the test set, leading to unreliable evaluation metrics. Third, stratified cross-validation ensures that each fold maintains the original class distribution, which is crucial for our highly imbalanced dataset where some categories contain as few as 5 samples (Children's Clothing at 1.0%). We report mean and standard deviation for accuracy, macro F1-score (averaging F1 across all categories equally to emphasize minority class performance), and weighted F1-score (weighting by category support) across all 5 folds.

6. Results and Discussion

Table 2 presents the classification results for all the models.

Model	Accuracy	Macro F1	Weighted F1
Traditional Machine Learning			
TF-IDF + SVM	0.419 ± 0.059	0.284 ± 0.087	0.355 ± 0.068
TF-IDF + LR	0.503 ± 0.026	0.417 ± 0.056	0.501 ± 0.033
Arabic Transformer Models			
AraBERT-v2	0.404 ± 0.066	0.162 ± 0.051	0.297 ± 0.079
CAMEL-BERT-mix	0.546 ± 0.035	0.303 ± 0.043	0.478 ± 0.039
MARBERT	0.586 ± 0.026	0.332 ± 0.019	0.522 ± 0.025
Large Language Model (GPT-4o-mini)			
Zero-Shot	0.064 ± 0.013	0.076 ± 0.025	0.079 ± 0.017
3-Shot	0.186 ± 0.007	0.172 ± 0.011	0.169 ± 0.014

Model	Accuracy	Macro F1	Weighted F1
TF-IDF + SVM	0.419 ± 0.059	0.284 ± 0.087	0.355 ± 0.068
TF-IDF + LR	0.503 ± 0.026	0.417 ± 0.056	0.501 ± 0.033
Arabic Transformer Models			
AraBERT-v2	0.404 ± 0.066	0.162 ± 0.051	0.297 ± 0.079
CAMEL-BERT-mix	0.546 ± 0.035	0.303 ± 0.043	0.478 ± 0.039
MARBERT	0.586 ± 0.026	0.332 ± 0.019	0.522 ± 0.025
Large Language Model (GPT-4o-mini)			
Zero-Shot	0.064 ± 0.013	0.076 ± 0.025	0.079 ± 0.017
3-Shot	0.186 ± 0.007	0.172 ± 0.011	0.169 ± 0.014

Table 2: Classification Results (5-Fold Cross-Validation, Mean ± Std). Best results per metric highlighted. Macro F1 treats all categories equally regardless of size.

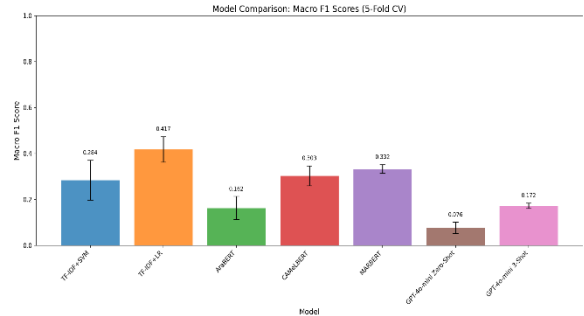


Figure 1: Model comparison showing macro F1 scores with error bars (standard deviation across 5 folds).

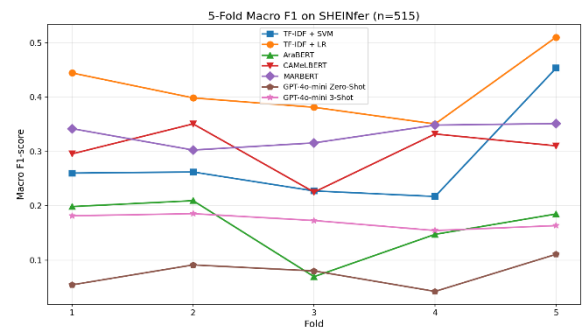


Figure 2: Macro F1-score variation across 5 folds for all models. TF-IDF+LR shows consistent superiority while transformer models exhibit higher variance.

6.1 Traditional ML Performance

As Figure 1 shows, TF-IDF with Logistic Regression achieves the best macro F1 (0.417 ± 0.056), substantially outperforming all other models on this metric. This result is notable given the simplicity of the approach and suggests that lexical features capture important category-specific vocabulary in Arabic e-commerce reviews. The strong performance of traditional ML methods on our small dataset is consistent with findings in the literature showing that simpler models often outperform deep learning approaches when training data is limited. TF-IDF+SVM achieves lower macro F1 (0.284 ± 0.087) with higher variance, indicating less stability across folds.

6.2 Transformer Model Analysis

MARBERT achieves the highest accuracy (0.586 ± 0.026) among all models, demonstrating the importance of pre-training on informal Arabic text for e-commerce review classification. Its pre-training on 1 billion Arabic tweets enables better understanding of dialectal expressions and colloquial language prevalent in SHEIN reviews. CAMeLBERT achieves the second-highest accuracy (0.546 ± 0.035), benefiting from its dialectal Arabic pre-training. However, both models show lower macro F1 scores (0.332 for MARBERT, 0.303 for CAMeLBERT) compared to TF-IDF+LR, indicating that while they excel at majority categories, they struggle with minority classes. AraBERT shows notably poor performance (0.162 macro F1), likely because its MSA-focused pre-training does not transfer well to dialectal e-commerce text. As shown in Figure 2, transformer models exhibit higher fold-to-fold variance compared to TF-IDF+LR, suggesting sensitivity to training data composition.

6.3 LLM Performance Analysis

GPT-4o-mini demonstrates surprisingly poor performance on this task. Zero-shot classification achieves only 0.064 ± 0.013 accuracy (near random for 11 classes), indicating that the model struggles to understand the implicit product references in dialectal Arabic reviews. The 3-shot setting improves performance to 0.186 ± 0.007 accuracy with better macro F1 (0.172 ± 0.011), representing a 3x improvement but still substantially below traditional approaches. Several factors may explain this: (1) Dialectal Arabic text with Saudi/Gulf expressions may be underrepresented in GPT-4o-mini's training data; (2) The implicit nature of product references requires domain-specific knowledge about e-commerce discourse; (3) The fine-grained 11-category classification is challenging without extensive examples; and (4) Cultural context in Arabic e-commerce reviews may not transfer from predominantly English training data.

6.4 Error Analysis

We conducted detailed error analysis on misclassified samples from the best macro F1 model (TF-IDF + Logistic Regression). The analysis reveals several systematic error patterns that illuminate the challenges of implicit product inference:

- Ambiguous Quality Descriptors:** Generic positive terms like "حلو" (nice), "جميل" (beautiful), "روعة" (wonderful), and "يجنن" (amazing) appear across all product categories and provide no discriminative signal. These reviews, constituting approximately 12% of our dataset (64 samples labeled as "Unclear"), represent an inherent limitation where even human annotators could not determine the product category. Analysis of per-category performance shows that "Unclear" reviews have the lowest average word count (5.8 words) compared to categories with clear product indicators like Shoes (15.7 words) or Pants/Jeans (17.4 words).
- Clothing Subcategory Confusion:** Reviews mentioning fabric quality ("خامة", "قماش") or fit ("مقاس") are generally correctly identified as clothing items but frequently confused among subcategories (Dress/Abaya, Blouse/Shirt, Pants/Jeans). The dialectal terms "يهفهف" (flows/flutter) and "بارد" (cool) typically indicate flowing garments like dresses but occasionally appear in reviews for other clothing items. Confusion matrix analysis reveals that 23% of Blouse/Shirt samples were misclassified as Dress/Abaya, and 18% of Skirt samples were confused with Dress/Abaya, reflecting the overlapping vocabulary used to describe these garment types.
- Category Overlap:** Products with overlapping characteristics cause systematic confusion. Accessories and Makeup/Beauty share descriptors related to appearance and aesthetics (31% confusion rate). Both Shoes and Clothing involve size/fit discussions using similar terminology ("مريح", "مقاس"). Home Products occasionally overlap with Accessories when reviews discuss decorative items.
- Minority Class Challenges:** Categories with fewer than 30 samples (Children's Clothing with 5, Skirt with 11, Underwear/Sportswear with 23) exhibit substantially lower F1-scores. Children's Clothing achieves only 0.12 F1-score despite having distinctive vocabulary like "طفل" (child), likely due to insufficient training examples. The extreme class imbalance contributes to the model's tendency to predict majority classes. Domain-specific terminology improves accuracy when present; reviews containing "بشرة" (skin) for Beauty or "مشي" (walking) for Shoes achieve

over 70% accuracy compared to 35% for reviews lacking such indicators.

7. Conclusion and Limitations

We introduced SHEINfer, a novel task and dataset for implicit product category inference from Arabic e-commerce reviews. Using 5-fold stratified cross-validation to ensure robust evaluation on our 515-sample dataset, our comprehensive experiments reveal that TF-IDF with Logistic Regression achieves the best macro F1 (0.417 ± 0.056), while MARBERT achieves the highest accuracy (0.586 ± 0.026) through better handling of majority categories and dialectal Arabic. Surprisingly, GPT-4o-mini performs poorly in both zero-shot (0.064 accuracy) and 3-shot (0.186 accuracy) settings, indicating that implicit product inference from dialectal Arabic remains challenging for general-purpose LLMs.

Our detailed error analysis reveals that ambiguous quality descriptors, clothing subcategory confusion, category overlap, and minority class challenges are the primary sources of classification errors. Future work should explore larger annotated datasets, domain-specific LLM fine-tuning, advanced prompting strategies, and multimodal approaches combining review text with product images. We will release our dataset and code upon paper acceptance to facilitate further research in Arabic e-commerce NLP.

Finally, our study has several limitations: (1) Dataset size is relatively small (515 samples) due to the resource-intensive dual annotation process. While we mitigate this through 5-fold stratified cross-validation, results may not fully generalize to larger datasets; (2) Category distribution is highly imbalanced, with Makeup/Beauty dominating (25.8%) while Children's Clothing has only 5 samples (1.0%); (3) The moderate inter-annotator agreement ($\kappa = 0.60$) reflects the inherent ambiguity in implicit product inference, which may affect gold label reliability; (4) LLM experiments were limited to GPT-4o-mini; other models (GPT-5, Claude, Gemini) may perform differently; (5) The dataset is specific to SHEIN's product catalog and Saudi/Gulf dialects, requiring validation on other platforms and dialects; (6) We did not explore more sophisticated prompting techniques (chain-of-thought, self-consistency) that might improve LLM performance.

8. References

- Abdul-Mageed, M., Elmadany, A., & Nagoudi, E. M. B. (2021). ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL), pages 7088-7105.
- Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2018). Deep Recurrent Neural Network vs. Support Vector Machine for Aspect-Based Sentiment Analysis of Arabic Hotels' Reviews. *Journal of Computational Science*, 27, 386-393.
- Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based Model for Arabic Language Understanding. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT), pages 9-15.
- Bin Safi, R. (2024). Arabic Reviews of SHEIN. Hugging Face Datasets. https://huggingface.co/datasets/Ruqiya/Arabic_Reviews_of_SHEIN
- Blume, A., Zalmout, N., Ji, H., & Li, X. (2023). Generative Models for Product Attribute Extraction. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 575-585.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., et al. (2020). Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems (NeurIPS), 33, pages 1877-1901.
- Çiftlikçi, M. S., Çakmak, Y., Kalaycı, T. A., Abut, F., Akay, M. F., & Kızıldağ, M. (2025). A New Large Language Model for Attribute Extraction in E-Commerce Product Categorization. *Electronics*, 14(10), 1930.
- Elnagar, A., Khalifa, Y. S., & Einea, A. (2018). Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications. In Intelligent Natural Language Processing: Trends and Applications, pages 35-52. Springer.
- Fang, C., Li, X., Fan, Z., Xu, J., Nag, K., Korpeoglu, E., Kumar, S., & Achan, K. (2024). LLM-Ensemble: Optimal Large Language Model Ensemble Method for E-commerce Product Attribute Value Extraction. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval.
- Habbat, N., Anoun, H., & Hassouni, L. (2023). Enhancing Arabic Sentiment Analysis in E-Commerce Reviews on Social Media Through a Stacked Ensemble Deep Learning Approach. *Mathematical Modelling of Engineering Problems*, 10(3), 867-876.
- Inoue, G., Alhafni, B., Baimber, N., Bouamor, H., & Habash, N. (2021). The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models. In Proceedings of the Sixth Arabic Natural Language Processing Workshop (WANLP), pages 92-104.
- Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1), 159-174.
- Lin, R., He, X., Feng, J., Zalmout, N., Liang, Y., Xiong, L., & Dong, X. L. (2021). PAM: Understanding Product Images in Cross Product Category Attribute Extraction. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pages 3192-3200.

- Wang, Q., Yang, L., Kanagal, B., Sanghai, S., Sivakumar, D., Shu, B., Yu, Z., & Elsas, J. (2020). Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 47-55.
- Yang, L., Wang, Q., Yu, Z., Kulkarni, A., Sanghai, S., Shu, B., Elsas, J., & Kanagal, B. (2022). MAVE: A Product Dataset for Multi-source Attribute Value Extraction. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pages 1256-1265.
- Zahera, H. M., & Sherif, M. A. (2020). ProBERT: Product Data Classification with Fine-tuned BERT Model. In Proceedings of the MIDAS Workshop at ESWC 2020.
- Zheng, G., Mukherjee, S., Dong, X. L., & Li, F. (2018). OpenTag: Open Attribute Value Extraction from Product Profiles. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1049-1058.
- Zhou, J., Liu, B., Acharya, J., Hong, Y., Lee, K.-C., & Wen, M. (2024). Enhanced E-Commerce Attribute Extraction: Innovating with Decorative Relation Correction and LLAMA 2.0-Based Annotation. In Proceedings of The Web Conference 2024.
- Zou, H. P., Samuel, V., Zhou, Y., Zhang, W., Fang, L., Song, Z., Yu, P. S., & Caragea, C. (2024a). ImplicitAVE: An Open-Source Dataset and Multimodal LLMs Benchmark for Implicit Attribute Value Extraction. In Findings of the Association for Computational Linguistics: ACL 2024, pages 338-354.
- Zou, H. P., Yu, G. H., Fan, Z., Bu, D., Liu, H., Dai, P., Jia, D., & Caragea, C. (2024b). EIVEN: Efficient Implicit Attribute Value Extraction using Multimodal LLM. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), pages 453-463.

NAJD-MT: High-Fidelity Saudi Najdi–English Training Data for Bidirectional Neural Machine Translation

Nour Qandos^{1*}, Samar Ahmed¹, Omer Nacar², Ahmad Alrabghi¹

Rahaf Al Hallay¹, Aya Hamod¹, Shaden Alsuhaim¹

¹NAMAA Community, Riyadh, Saudi Arabia

²Tuwaiq Academy, Riyadh, Saudi Arabia

*nooramerg0@gmail.com

Abstract

Dialectal Arabic remains significantly underrepresented in parallel resources for direct machine translation with English, particularly for regional varieties such as Saudi Najdi Arabic. In this work, we introduce **NAJD-MT**, a systematically constructed Saudi Najdi→English parallel corpus designed for training bidirectional neural machine translation models. Starting from the Saudi Arabic Dialectal Annotated (SADA) dataset, we generate English translations using GPT-4.1 and subsequently apply cross-lingual embedding-based cosine similarity filtering to improve semantic alignment and reduce translation noise. We analyze the impact of varying semantic similarity thresholds on corpus size and downstream translation performance. Using the constructed datasets, we train and evaluate multiple Transformer-based models, including NLLB-200, OPUS-MT, mBART, and AraT5v2, in both Najdi→English and English→Najdi directions. Experimental results demonstrate that stricter semantic filtering ($\text{cosine} \geq 0.7$) consistently improves translation quality despite reducing dataset size, highlighting that data purity plays a critical role in dialectal machine translation training. Our findings provide a reproducible framework for constructing high-fidelity dialect English parallel corpora and emphasize the importance of semantic alignment filtering in low-resource dialectal settings.

Keywords: Dialectal Arabic, Machine translation, Semantic Similarity Filtering, Cross-Lingual Embeddings, Multilingual Transformers

1. Introduction

Recent advances in multilingual neural machine translation (NMT) have significantly improved translation quality across high-resource language pairs (Tang et al., 2020; Costa-Jussà et al., 2022). However, dialectal Arabic remains substantially underrepresented in parallel corpora, particularly for direct translation between regional dialects and English. While Modern Standard Arabic (MSA) benefits from large-scale datasets and extensive pre-training coverage, regional varieties such as Najdi Arabic continue to face severe data scarcity and limited standardized resources.

Dialectal Arabic differs from MSA at multiple linguistic levels, including phonology, morphology, syntax, and lexicon (Habash, 2010). These divergences are not merely surface-level variations; they often involve distinct morphosyntactic constructions and culturally embedded expressions. As a result, models trained predominantly on MSA frequently struggle when applied to dialectal input, leading to semantic drift and reduced robustness in conversational translation scenarios.

Previous research has explored pivot-based approaches for dialectal machine translation, typically translating dialectal Arabic to MSA before translating to English (Salloum and Habash, 2013). While pivoting can mitigate data scarcity, it may introduce cumulative translation errors and obscure dialect-specific nuances. Direct dialect-to-English trans-

lation has been shown to be preferable when sufficient parallel data is available, yet constructing such corpora remains costly and labor-intensive.

More recently, large language models (LLMs) have been employed for synthetic data generation and corpus augmentation in low-resource settings (Wang et al., 2023). However, automatically generated translations may introduce semantic inconsistencies or hallucinations, particularly when applied to dialectal input. Ensuring high-quality alignment therefore requires additional filtering mechanisms. Our main contributions are:

- We propose a scalable pipeline for constructing semantically aligned Najdi→English parallel data using large language model translation and embedding-based filtering.
- We analyze the impact of semantic similarity thresholds on corpus size and downstream NMT performance.
- We provide empirical evidence that stricter semantic filtering improves bidirectional dialectal translation quality.

2. Related Work

Neural machine translation for Arabic has achieved strong performance for Modern Standard Arabic (MSA), largely due to the availability of large parallel corpora and multilingual pretraining (Tang et al.,

2020; Costa-Jussà et al., 2022). However, dialectal Arabic remains under-resourced and presents additional challenges stemming from lexical variation, morphosyntactic divergence, and orthographic inconsistency (Habash, 2010; Zaidan and Callison-Burch, 2014).

Early approaches to dialect-to-English translation relied on pivoting through MSA (Salloom and Habash, 2013), where dialectal input is first normalized or translated into MSA before translating to English. Although pivoting reduces data requirements, it may introduce cascading errors and weaken dialect-specific semantic preservation. More recent studies emphasize the importance of direct dialect-to-English translation when parallel resources are available, as this approach better preserves colloquial and culturally embedded expressions (Bouamor et al., 2018).

Despite these advances, Saudi dialects particularly Najdi remain sparsely represented in publicly available parallel datasets. The SADA corpus (Alharbi et al., 2024) provides dialectal speech transcripts but does not include aligned English translations, leaving a gap for direct dialect-English MT training.

Synthetic data augmentation has become a common strategy in low-resource machine translation. Back-translation (Sennrich et al., 2016) and self-training approaches have demonstrated that synthetic parallel data can significantly improve NMT performance. More recently, large language models (LLMs) have been used to generate synthetic instruction-following and translation data (Wang et al., 2023), offering scalable solutions for data construction.

However, automatically generated translations may introduce semantic drift, hallucination, or stylistic inconsistencies, particularly in dialectal contexts where linguistic variability is high. Consequently, quality control mechanisms are necessary to ensure that synthetic parallel data maintains semantic fidelity.

Cross-lingual sentence embeddings enable semantic comparison across languages by mapping text into a shared representation space (Reimers and Gurevych, 2019; Gao et al., 2021). Such representations have been widely used for bitext mining and parallel sentence extraction (Schwenk et al., 2021; Artetxe and Schwenk, 2019). Cosine similarity thresholds are commonly employed to filter noisy sentence pairs and improve corpus quality.

In contrast to large-scale web-mined bitext extraction, our work applies embedding-based cosine similarity filtering to synthetic dialect-generated translations, evaluating how semantic filtering thresholds impact downstream NMT training performance.

While prior research has explored dialectal

MT, pivot-based strategies, synthetic data augmentation, and embedding-based bitext mining, limited work has examined systematic pipelines for constructing semantically aligned Saudi dialect-English training corpora. Our contribution lies in integrating LLM-based translation with cross-lingual semantic filtering for Najdi Arabic and empirically analyzing how alignment thresholds influence bidirectional NMT training outcomes.

3. Methodology

This section describes the construction of the NAJD-MT corpus and the training setup used to evaluate its impact on bidirectional Najdi→English neural machine translation (NMT). The overall pipeline is illustrated in Figure 1. The process consists of three stages: (1) source preparation and cleaning, (2) parallel data construction with semantic alignment filtering, and (3) bidirectional NMT training and evaluation.

3.1. Source Dataset and Preprocessing

We construct our corpus from the Saudi Arabic Dialectal Annotated (SADA) dataset (Alharbi et al., 2024), a speech-based resource containing transcribed Saudi dialect conversations. Since SADA includes both Najdi and Hejazi varieties, we restrict our experiments to the Najdi subset to reduce dialectal variability and promote homogeneity during model training.

Because SADA is derived from speech-to-text (STT) transcription, it contains noise artifacts, segmentation errors, and occasional transcription markers. We apply the following preprocessing steps: Removal of single-character or non-linguistic fragments, Deduplication of identical utterances, and Removal of STT artifacts such as the token "غير واضح". After preprocessing, the cleaned Najdi corpus serves as the source-side data for parallel corpus construction.

3.2. LLM-Based Translation

To construct the English side of the parallel corpus, we translate the cleaned Najdi sentences using GPT-4.1. Prior to full translation, we conducted a pilot comparison across multiple large language models (GPT-4.1, LLaMA-4-Maverick, Gemini-2.5-Pro, Claude-Sonnet-4, and Command-R-Plus) on 20 representative samples. Models were assessed across four criteria: hallucination rate, named entity preservation, semantic faithfulness, and dialectal term capture. Evaluation was conducted via blind human assessment by a bilingual Najdi Arabic-English expert and automated LLM-as-a-judge scoring using chatGPT. GPT-4.1

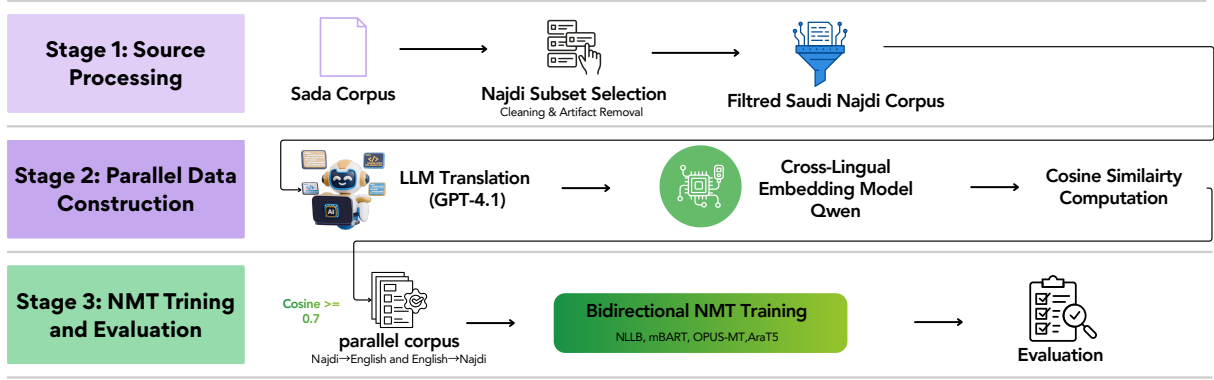


Figure 1: Methodology Pipeline

Stage	Remaining	Removed
Original	69,978	0
Single-character removal	69,945	33
Duplicate removal	69,663	282
Artifact removal	68,158	1,505
Embedding filter ($\tau \geq 0.6$)	54,280	15,698
Embedding filter ($\tau \geq 0.7$)	30,421	39,557

Table 1: Dataset size after each filtering stage.

demonstrated stronger semantic preservation and fewer hallucinations during manual inspection and was therefore selected for full-dataset translation.

The output of this stage consists of synthetic Najdi→English sentence pairs.

3.3. Semantic Similarity Filtering

Although large language models provide high-quality translations, automatic generation may introduce semantic drift, contextual expansion, or partial meaning loss. To ensure alignment fidelity, we apply cross-lingual embedding-based filtering using **Qwen3-Embedding-8B** (Zhang et al., 2025), a multilingual embedding model producing 4096-dimensional representations trained on large-scale multilingual corpora. Table 1 summarizes the corpus size after each preprocessing and filtering stage.

Let $x_{ar} \in R^d$ and $x_{en} \in R^d$ denote the cross-lingual sentence embeddings of the Arabic source sentence and its English translation, respectively. The cosine similarity between the two embeddings is computed as:

$$\text{sim}(x_{ar}, x_{en}) = \frac{x_{ar}^\top x_{en}}{\|x_{ar}\|_2 \|x_{en}\|_2} \quad (1)$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm. A sentence pair (x_{ar}, x_{en}) is retained in the filtered dataset if:

$$\text{sim}(x_{ar}, x_{en}) \geq \tau \quad (2)$$

where $\tau \in \{0.6, 0.7\}$ is the semantic similarity threshold.

3.4. Human Validation

To validate the reliability of embedding-based filtering, we conduct targeted manual inspection. First, all sentence pairs with cosine similarity ≤ 0.3 (158 samples) are reviewed. The majority exhibit severe semantic drift, confirming that low similarity correlates with misalignment.

Second, we randomly sample 170 sentence pairs with cosine similarity ≥ 0.7 . None contain critical mistranslations, and only minor deviations are observed in a small subset.

Validation is performed by two bilingual evaluators with native Najdi Arabic proficiency and advanced English competency, who independently assessed translation correctness and flagged dialectal inaccuracies, with disagreements resolved through discussion.

3.5. Dataset Splits

For each filtered dataset ($\tau = 0.6$ and $\tau = 0.7$), we apply an 80/20 split into training and validation sets. Splitting is performed after filtering to prevent leakage of low-quality pairs into evaluation data. No sentence overlap exists across splits.

3.6. Neural Machine Translation Models

To evaluate the effectiveness of the constructed corpus, we fine-tune multiple Transformer-based models:

- mBART (Tang et al., 2020)

- NLLB-200-distilled (Costa-jussà et al., 2022)
- OPUS-MT (Tiedemann and Thottingal, 2020)
- AraT5v2 (Elmadany et al., 2023)

We train models in both Najdi→English and English→Najdi directions.

3.7. Training Setup

All models are fine-tuned using the HuggingFace Transformers framework. Training is conducted on NVIDIA RTX 5090 GPUs with the following hyperparameters: Learning rate: 2×10^{-5} , Linear decay scheduler, Batch size: 64, Maximum epochs: 30, Early stopping based on validation BLEU. Evaluation is performed using BLEU to ensure standardized and reproducible scoring.

4. Experiments and Results

We evaluate bidirectional Najdi→English translation using BLEU (Papineni et al., 2002) as the primary automatic metric. Additionally, we report BERTScore (Zhang et al., 2020) for the top-performing model in each translation direction to provide complementary semantic-level evaluation. All scores are reported on the validation set.

4.1. Najdi → English

Table 2 presents BLEU scores for translation from Najdi Arabic to English under two semantic filtering thresholds ($\tau = 0.6$ and $\tau = 0.7$).

Across all models, training on the more strictly filtered dataset ($\tau = 0.7$) consistently yields higher BLEU scores despite the reduction in corpus size. NLLB-200 achieves the highest BLEU score (30.60) at $\tau = 0.7$, followed closely by OPUS-MT (29.10) and mBART (28.40). BERTScore evaluation of NLLB-200 at $\tau = 0.7$ further supports this result (P: 0.9406, R: 0.9370, F1: 0.9387), indicating strong semantic adequacy beyond surface n-gram overlap. AraT5v2 shows slightly lower performance but remains competitive. These results suggest that stricter semantic filtering improves translation quality in the Najdi→English direction.

4.2. English → Najdi

Table 3 reports results for English→Najdi translation. Overall BLEU scores are substantially lower than in the reverse direction, reflecting the greater difficulty of generating dialectal output. mBART achieves the highest BLEU score (15.60) at $\tau = 0.7$, followed closely by OPUS-MT (15.10). Similar to the previous direction, models trained on the $\tau = 0.7$ dataset outperform those trained on $\tau = 0.6$. BERTScore evaluation of mBART at $\tau = 0.7$ (P:

Model	$\tau = 0.6$	$\tau = 0.7$
AraT5v2	25.53	26.80
mBART	26.0	28.40
OPUS-MT	25.83	29.10
NLLB-200	27.60	30.60

Table 2: Najdi→English BLEU scores.

Model	$\tau = 0.6$	$\tau = 0.7$
AraT5v2	12.40	13.50
mBART	14.43	15.60
OPUS-MT	13.70	15.10

Table 3: English→Najdi BLEU scores.

0.9122, R: 0.9101, F1: 0.9111) further reveals that despite low BLEU scores, the model captures substantial semantic meaning even when exact dialectal surface forms are not reproduced. The relatively small performance differences across models suggest that dialect generation remains challenging and may be constrained by limited dialectal representation in multilingual pretraining.

5. Discussion

Our experiments highlight two central findings. First, stricter semantic filtering consistently improves translation performance across all evaluated models and both translation directions. Despite reducing the corpus size from 54,280 to 30,421 sentence pairs, increasing the cosine similarity threshold from $\tau = 0.6$ to $\tau = 0.7$ yields consistent BLEU improvements. This suggests that, in dialectal settings, semantic alignment quality may outweigh data volume. Noisy or weakly aligned sentence pairs appear to introduce greater harm during training than the benefit gained from additional quantity.

Second, a clear directional asymmetry is observed. Najdi→The English translation achieves substantially higher BLEU scores than the reverse direction. This asymmetry likely reflects the broader representation of English in multilingual pretraining corpora, as well as the relative difficulty of generating dialectal output. While multilingual NMT models are generally exposed to English during pretraining, dialectal Arabic particularly Najdi remains sparsely represented. Consequently, generating dialect-specific morphology, vocabulary, and colloquial constructions presents a greater challenge than translating dialectal input into standardized English.

Across models, NLLB-200 demonstrates the strongest performance in the Najdi→English direction, suggesting that large-scale multilingual pre-

training combined with distillation remains effective even for underrepresented dialectal input. However, performance differences between models are relatively modest, particularly in the English→Najdi direction, indicating that architectural variation alone is insufficient to overcome dialectal data scarcity.

Overall, the results reinforce the importance of corpus construction methodology in low-resource dialectal machine translation. The consistent gains obtained through embedding-based filtering demonstrate that semantic similarity can serve as an effective quality-control mechanism for synthetic parallel data.

6. Error Analysis

To systematically investigate the model's limitations in English→Najdi direction, an error analysis was conducted using a representative sample of outputs generated by the best-performing model in this translation direction. Overall, the model demonstrates strong performance in translating short, everyday, and relatively simple sentences. However, several recurring issues were observed, particularly related to gender agreement, where masculine and feminine forms are occasionally confused (e.g., translating possessive forms incorrectly). The model also shows clear limitations when handling longer and more complex sentences. This may be attributed to the loss of semantic nuances and contextual information during the translation process between English and Najdi. Notably, the model's performance improves when the source sentence is well-structured and unambiguous.

Following the same methodology, an error analysis was performed on samples generated by the best-performing model in the Najdi→English direction. One prominent issue is the repetition of certain words, especially named entities, where the model tends to repeat frequently recognized tokens (e.g., proper names such as "Fatimah") at the expense of preserving the overall context. Despite this, the model demonstrates relatively better handling of longer texts, producing translations that are moderately coherent. This improvement may be due to the relative clarity and structural richness of the source Najdi Arabic text compared to the generated English output.

7. Conclusion

In this work, we introduced Najdi-MT, a semantically aligned Najdi→English parallel corpus constructed through large language model translation and cross-lingual embedding-based filtering. We demonstrated that applying stricter semantic similarity thresholds improves downstream bidirectional neural machine translation performance, even at

the cost of reduced dataset size. Our experiments show that data quality plays a critical role in dialectal machine translation, particularly in low-resource settings. While multilingual models such as NLLB-200 and mBART perform competitively, improvements remain constrained by limited dialectal representation during pretraining. The observed directional asymmetry further highlights the challenges of generating dialectal Arabic compared to translating it into English. These findings suggest that careful corpus construction and alignment filtering are essential for advancing dialect-aware machine translation systems. Future work will extend Najdi-MT to additional Saudi dialects, incorporate broader domain coverage.

8. References

- Sadeen Alharbi, Areeb Alowisheq, Zoltán Tüske, Kareem Darwish, Abdullah Alrajeh, Abdulmajeed Alrowithi, Aljawharah Bin Tamran, Asma Ibrahim, Raghad Aloraini, Raneem Alnajim, et al. 2024. Sada: Saudi audio dataset for arabic. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10286–10290. IEEE.
- Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3197–3203.
- Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghoulani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhil Eryani, Alexander Erdmann, et al. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.
- Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil

- Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. [No language left behind: Scaling human-centered machine translation](#). *arXiv preprint arXiv:2207.04672*.
- AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. [Octopus: A multitask model and toolkit for Arabic natural language generation](#). In *Proceedings of Arabic-NLP 2023*, pages 232–243, Singapore (Hybrid). Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6894–6910.
- Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.
- 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. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.
- Wael Salloum and Nizar Habash. 2013. Dialectal arabic to english machine translation: Pivoting through modern standard arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 348–358.
- Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. Ccmatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 86–96.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. [Multilingual translation with extensible multilingual pretraining and fine-tuning](#).
- Jörg Tiedemann and Santhosh Thottingal. 2020. [Opus-mt — building open translation services for the world](#). In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.
- Omar F Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#).
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Parsing Arabic Dialects Revisited: New Benchmarks, Models, and Insights

Ahmed Elshabrawy,^{*,†} Go Inoue,^{†,‡} Muhammed AbuOdeh,[‡] Nizar Habash^{†,‡}

[†]Computational Approaches to Modeling Language (CAMEL) Lab,
New York University Abu Dhabi (NYUAD)

[‡]Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

{ahmed.elshabrawy, go.inoue, muhammed.abuodeh}@mbzuai.ac.ae, nizar.habash@nyu.edu

Abstract

Parsing dialectal Arabic remains underexplored, with limited progress over the past two decades. Existing Modern Standard Arabic (MSA) parsers perform poorly on dialectal data, motivating the need for dialect-specific approaches. We revisit this task using modern neural models and present new results on Egyptian and Gulf Arabic dependency parsing. We demonstrate that even small amounts of dialectal training data yield substantial improvements in parsing accuracy. Our contributions include: (1) introducing a new annotated dataset for Gulf Arabic, (2) releasing a state-of-the-art multi-variety Arabic parser, and (3) employing dialect identification as a diagnostic tool to better understand how training data affects parsing performance across dialects and test sets.

Keywords: Arabic Dependency Parsing, Dialectal Arabic, Gulf Arabic Dataset, Egyptian Arabic, Language Resources, Annotated Corpora, Dialect Identification, Open-Source NLP Tools, Modern Standard Arabic (MSA)

1. Introduction

Arabic dialects, such as Egyptian (EGY), Gulf (GLF), Levantine, and Maghrebi, are the dominant spoken varieties across the Arab world. They pervade informal communication, social media, and everyday discourse, yet remain underrepresented in computational linguistics (Darwish et al., 2021). In contrast, Modern Standard Arabic (MSA) has long benefited from extensive linguistic resources and treebanks (Maamouri et al., 2004; Habash and Roth, 2009; Taji et al., 2017; Habash et al., 2022). As Table 1 shows, even strong MSA parsers like CamelParser2.0 (Elshabrawy et al., 2023) degrade significantly on dialectal input, underscoring the need for dialect-focused approaches.

While large language models (LLMs) raise questions about the role of traditional parsing, syntactic analysis remains crucial for tasks requiring interpretability and structure, especially in education, grammar-aware tools, and low-resource settings (Guo et al., 2024). Parsing also offers reliable scaffolding for downstream tasks such as error detection and information extraction, where LLMs may lack consistency or transparency (Kanayama et al., 2024; Li et al., 2025; Fan et al., 2025).

This paper revisits Arabic dialectal parsing, a task largely neglected for over two decades (Chiang et al., 2006). We adopt a modern neural dependency parsing architecture to build a state-of-the-art open-source parser for multiple variants of Arabic.

Our main contributions are: (a) the introduction of a **new annotated Gulf Arabic dependency treebank**, enabling the first parsing results for this

System	MSA	EGY	GLF	AVG
CamelParser2.0	87.5	73.3	72.7	77.8
Ours	87.2	84.2	80.3	83.9

Table 1: Labeled Attachment Score (LAS) performance of the MSA-trained CamelParser2.0 and of our system on MSA, EGY, GLF test sets.

dialect; (b) the release of a **state-of-the-art multi-variety Arabic parser (MSA, EGY, GLF)** (Table 1); and (c) the use of **dialect identification as a diagnostic tool** to assess parsing complexity and data impact.¹

2. Related Work

2.1. Arabic Treebanks

A number of Arabic treebanks have been developed, differing in size, syntactic formalism, and genre coverage. Among these, the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) has been the primary resource for MSA syntactic analysis. The PATB consists mainly of newswire and web texts. Although originally annotated using a phrase-structure representation, it has been converted into several dependency formalisms (Smrž et al., 2002; Habash and Roth, 2009; Taji et al., 2017).

Building on PATB and related efforts, several MSA dependency treebanks have been developed.

¹We make our code and data available at https://github.com/CAMEL-Lab/camel_parser_dialects.

* Equal contribution.

PADT was the first Arabic dependency treebank and employs a multi-layered annotation scheme capturing functional morphology, analytical dependency syntax, and tectogrammatical representations of meaning (Smrž et al., 2002). The Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009) introduced a streamlined dependency representation inspired by traditional Arabic grammar. The Quranic Arabic Corpus (Dukes and Buckwalter, 2010) provided a hybrid dependency–constituency annotation of Quranic scripture. The Arabic Poetry Treebank (ArPoT) (Al-Ghamdi et al., 2021) applies the CATiB formalism, with extensions, to Classical Arabic poetry. The *i3rab* treebank (Halabi et al., 2021) adopts a dependency representation closely aligned with traditional Arabic grammatical theory. The Camel Treebank (CamelTB) (Habash et al., 2022) has further expanded MSA dependency resources across multiple genres and time periods.

In addition to MSA, several dialectal treebanks have been developed. The Linguistic Data Consortium (LDC) released dialectal resources derived from conversational transcripts, including the Levantine Arabic Treebank (LATB) (Maamouri et al., 2006) and the Egyptian Arabic Treebank (ARZTB) (Maamouri et al., 2014). More recently, a 22K-token Algerian Arabic treebank was introduced by Seddah et al. (2020).

2.2. Parsing Arabic and its Dialects

Parsing efforts in Arabic have predominantly focused on MSA, largely due to the presence of rich annotated datasets. Early work on MSA parsing includes Habash and Roth (2009), where they built a transition-based dependency parser using Malt-Parser (Nivre et al., 2006). This work was extended in subsequent systems with improved features, such as those by Marton et al. (2013) and Shahrour et al. (2016). Recent work on neural Arabic dependency parsing has explored a variety of modeling approaches, including multi-task easy-first dependency parsing (Kankanampati et al., 2020), dependency parsing as a sequence labeling task (Al-Ghamdi et al., 2023), and graph-based biaffine dependency parsing (Elshabrawy et al., 2023).

Dialectal Arabic parsing, however, has received limited attention. Early pioneering work by Chiang et al. (2006) explored a treebank transduction approach, leveraging MSA resources for parsing Levantine Arabic. Muller et al. (2020) investigated the transferability of multilingual language models to Algerian Arabic for dependency parsing. Abo Mokh et al. (2024) evaluated the performance of an MSA-trained parser on a small set of sentences in Gulf, Levantine, Egyptian, and Maghrebi dialects. In this work, we present empirical studies on building a state-of-the-art dialectal dependency parser for Egyptian and Gulf Arabic.

In this work, we adopt the CATiB dependency representation across MSA, EGY, and GLF for consistency. For MSA, we use CamelTB and train on both the CATiB-converted PATB and CamelTB data. For EGY, we automatically convert ARZTB into CATiB style. For GLF, we introduce the first dependency treebank for the dialect, consisting of over 25K tokens, manually annotated following CATiB guidelines with minor extensions. All evaluations are conducted in the CATiB framework.

2.3. Dialect Identification

Dialect Identification (DID) is the task of predicting the dialectal variety of a given speech or text segment (Etman and Beex, 2015). There has been a growing interest in DID, reflected in a series of shared evaluation campaigns, including MADAR and NADI (Bouamor et al., 2019; Abdul-Mageed et al., 2021, 2024), and the development of various datasets and tools (Zaidan and Callison-Burch, 2011; Bouamor et al., 2014; Salama et al., 2014; Alsarsour et al., 2018; Abu Kwaik et al., 2018; Zaghoulani and Charfi, 2018; Salameh et al., 2018; Bouamor et al., 2019; Abdelali et al., 2021; Baimukan et al., 2022).

DID has been shown to improve downstream NLP performance by enabling dialect-aware system selection or adaptation, including in machine translation (Salloum et al., 2014) and morphological tagging (Obeid et al., 2022). In this work, we adopt DID not as an end task, but as an analytical tool to obtain finer-grained evaluation subsets, allowing us to better understand cross-variety parsing behavior and training data effects.

3. Arabic Syntax Representation

We follow the **Columbia Arabic Treebank (CATiB)** annotation scheme (Habash et al., 2009, 2022). Grounded in traditional Arabic grammar yet deliberately simplified, CATiB is intuitive for native speakers and well-suited for consistent annotation across Arabic varieties. Figures 1 and 2 are examples of CATiB annotated trees. Next is a very compact summary of the CATiB representation.

In terms of **tokenization**, CATiB segments all clitics except the definite article *Al*+ ‘the’, and normalizes basewords to their uncliticized form.

CATiB uses six **part-of-speech** tags: *NOM* (common nominals), *PROP* (proper nouns), *VRB* (active verbs), *VRB-PASS* (passive verbs), *PRT* (particles, including prepositions and conjunctions), and *PNX* (punctuation). This compact tagset supports efficient annotation while maintaining syntactic adequacy.

As for **dependency relations**, CATiB uses eight: *SBJ* (subjects of verbs and topics of simple nominal

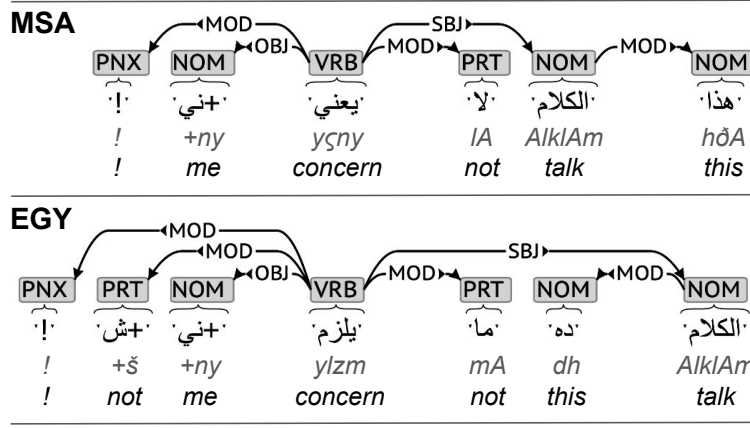


Figure 1: Parallel MSA and EGY CATiB dependency trees for ‘this talk does not concern me.’ Words are shown with transliteration and gloss.

Dataset	#Sentences				#Tree Tokens			
	ALL	TRAIN	DEV	TEST	ALL	TRAIN	DEV	TEST
PATB (Maamouri et al., 2004)	19,738	15,789	1,986	1,963	738,889	590,819	73,945	74,125
CamelTB (Habash et al., 2022)	13,337	9,397	2,022	1,918	241,910	171,735	35,418	34,757
ARZTB (Maamouri et al., 2014)	31,688	24,428	3,542	3,718	508,548	393,193	57,360	57,995
CamelTB-Gumar (Ours)	2,881	2,021	435	425	25,516	17,648	4,112	3,756
Total	67,644	51,635	7,985	8,024	1,514,863	1,173,395	170,835	170,633

Table 2: Dataset statistics for all treebanks used in this study. For each dataset, we report the total number of sentences and tree tokens (ALL), together with their counts in the TRAIN, DEV, and TEST splits. Data splits follow Diab et al. (2013) for PATB and ARZTB and Habash et al. (2022) for CamelTB.

sentences), *OBJ* (objects of verbs, prepositions, or deverbal nouns), *TPC* (topics of complex nominal sentences with explicit pronominal reference), *PRD* (predicative complements in extended copular constructions), *IDF* (the *idafa* possessive construction), *TMZ* (the *tamyiz* specification construction), **MOD** (general modification of verbs or nominals), and — (flat constructions, e.g., multiword proper names).

Figure 1 shows parallel MSA and EGY CATiB analyses. Both have five orthographic words, but clitic segmentation produces six tokens in MSA and seven in EGY. The trees show structural similarities, such as SVO order (though MSA often uses VSO), preverbal negation, and direct object pronominal clitics. They also highlight differences, including EGY suffix negation (+ش ‘not’) and postposed demonstratives (ده ‘this’).

4. Arabic Treebanks

In this paper, we make use of four Arabic treebanks: two MSA, and two dialectal (one EGY and one GLF). The latter (CamelTB-Gumar) is a new treebank we introduce as part of this work and make publicly available. Table 2 shows tree token counts across splits for all treebanks. Two of the treebanks (one

MSA and EGY) were converted from constituency representation to CATiB dependency; and two were created directly in CATiB dependency representation. We discuss each of these resources next, and finish the section with a comparison of lexical and structural differences among them.

4.1. PATB (MSA Newswire)

The Penn Arabic Treebank (PATB) introduced earlier in Section 2 is the largest Arabic treebank to date.² PATB focuses on MSA for newswire and was created in constituency representation (Maamouri et al., 2004). Following Habash and Roth (2009), we use an automatically converted version to CATiB created with the Arabic-C2D utility.³ We use the data splits recommended by Diab et al. (2013). The same conversion and splits are used in CamelParser 1.0 and 2.0 (Shahrour et al., 2016; Elshabrawy et al., 2023). In this paper we only use PATB as part of the training.

²LDC2010T13, LDC2011T09, LDC2010T08

³Arabic-C2D GitHub repository.

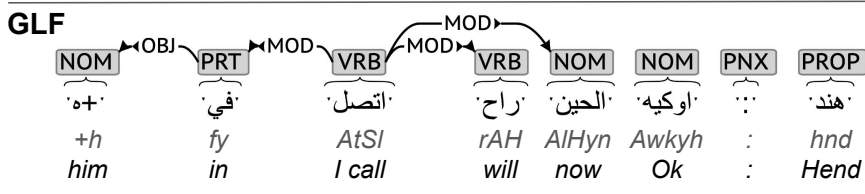


Figure 2: GLF CATiB tree for ‘Hend: OK, I’ll call him now.’ Words are shown with transliteration and gloss.

4.2. CamelTB (MSA Mixed Genres)

The Camel Treebank (CamelTB) includes 13 sub-corpora comprising selections of texts from pre-Islamic poetry to social media online commentaries, and covering a range of genres from religious and philosophical texts to news, novels, and student essays (Habash et al., 2022). The CamelTB was annotated directly in CATiB representation. We use the data split recommended by Habash et al. (2022). The CamelTB Train is used with PATB Train as the MSA Train baseline, as used in CamelParser2.0. The CamelTB Dev and Test sets are our only MSA test sets. We chose to focus on CamelTB as MSA for testing because of its genre diversity.

4.3. ARZTB (EGY Forums & Chat)

For Egyptian Arabic, we use the Egyptian Arabic Treebank (ARZTB) (Maamouri et al., 2014), a constituency representation treebank containing data from discussion forums, text messaging, and chat.⁴ We converted to the CATiB formalism following the approach of Habash and Roth (2009) discussed above for PATB.³ We use the data splits recommended by Diab et al. (2013). We make our Arabic-C2D conversion map publicly available.

4.4. CamelTB-Gumar (GLF Internet Novels)

We introduce the Gulf Treebank (CamelTB-Gumar), the first dependency treebank for Gulf Arabic, annotated in the Columbia Arabic Treebank (CATiB) scheme (Habash and Roth, 2009), and following the updated guidelines used in the Camel Treebank (Habash et al., 2022). The CamelTB-Gumar data comprises the development set of the 200K-word morphologically annotated portion of the Gumar Corpus (Khalifa et al., 2016, 2018). We define our own splits for CamelTB-Gumar following Diab et al. (2013)’s guidelines. We make the treebank and splits publicly available.

We leverage the existing gold tokenization and POS tags to create syntactic annotations, starting from automatic parses generated by CamelParser1.0 (Shahrour et al., 2016), trained on PATB

	MSA	EGY	GLF
PATB (MSA)	91.2 / 92.6	82.3 / 96.4	65.5 / 74.1
CamelTB (MSA)		85.7 / 97.4	72.0 / 87.9
ARZTB (EGY)			75.7 / 83.9

Table 3: Pairwise cosine similarity (%) across treebanks (**Lexical** / **Structural**). Columns correspond to reference treebanks: MSA (CamelTB), EGY (ARZTB), and GLF (CamelTB-Gumar).

data. Annotation was carried out by a team of native speakers with extensive experience in linguistic annotation, hired through a professional annotation firm and fairly compensated. An expert Arabic native speaker linguist manually reviewed and corrected 125 sentences (1,218 tokens) from the training data, representing ~5% of the full corpus. The resulting agreement scores were very high: POS accuracy reached 99.9%, while syntactic scores were 99.0% UAS, 99.2% LS, and 98.9% LAS. These results indicate that the annotations are of excellent quality.

The CamelTB-Gumar differs markedly from other Arabic treebanks due to its genre: conversational internet novels. The text contains frequent fragments, dialogue formatted as (*Name: Speech*), ellipsis, elided vocatives, missing punctuation, and interrupted phrases, making it syntactically more challenging and distinct from formal newswire treebanks. Figure 2 illustrates a typical example, where a minimally punctuated sentence is analyzed as four fragments under a single root.

4.5. Treebank Analysis

We compare the four treebanks in terms of lexical and structural similarity. Table 3 reports pairwise cosine similarity over lexical distributions and structural dependency bigrams.

Lexical Similarity The two MSA treebanks (PATB and CamelTB) are the most similar lexically (91.2%), while EGY treebank shows moderate similarity to MSA treebanks (82–86%). The GLF treebank is the most divergent variety, with similarity to MSA as low as 65.5%. Overall, the results indicate

⁴LDC2018T23

a lexical gradient (MSA > EGY > GLF), suggesting increasing difficulty for cross-treebank transfer involving GLF.

Clitic distributions further illustrate these lexical differences. Core clitics such as $w+$ ‘and’, $h+$ ‘his’, $b+$ ‘with/by’, and $l+$ ‘to/for’ are frequent across all treebanks. Dialect-specific clitics appear prominently in ArzTB, including the future marker $H+$ ‘will’ and the negation suffix $š+$ ‘not’, which are absent from MSA and GLF. Conversely, GLF exhibits distinctive forms such as $ج++j$ ‘you/your [f.s.]’ and $ش+š$ ‘what’ that do not occur in the other datasets.

Structural Similarity For structural comparison, we compute cosine similarity over the frequencies of 235 dependency bigram types (e.g., *VRB-SBJ-NOM*) in the union of all treebanks. Table 3 shows that structural similarity is generally higher than lexical similarity. ARZTB is highly similar to both MSA treebanks (96–97%), even exceeding the similarity between PATB and CamelTB. This may reflect the broader genre and stylistic diversity of CamelTB and ARZTB, compared to the more homogeneous PATB (newswire) and CamelTB-Gumar (internet novels). GLF remains the most distinct variety, though its structural distance (74–88%) is less pronounced than its lexical distance. Overall, the results suggest that cross-treebank variation is driven more by lexical differences than by major syntactic restructuring.

Frequent structural tuples further support this pattern. Core configurations such as *PRT-OBJ-NOM*, *NOM-IDF-NOM*, and *VRB-OBJ-NOM* are almost categorically head-first across all treebanks, indicating stable word order. Greater variation appears in subject and modifier relations: for example, *VRB-SBJ-NOM* is predominantly head-first in MSA (80–85%) but much less so in the dialectal treebanks (36–39%). Similarly, nominal verb modifier relations (e.g., *VRB-MOD-NOM*) show stronger head-first tendencies in MSA treebanks than in the dialectal treebanks. These differences point to subtle shifts in syntactic realization in the dialects, despite overall structural similarity.

5. Experimental Setup

In this section, we describe the data, model configurations, and evaluation protocol used to study multi-variety Arabic parsing.

5.1. Data

Evaluation Data Table 4 shows the development and test splits we use in our experiment. For MSA, we use the CamelTB development and test sets and

exclude PATB because CamelTB is more diverse (genre-wise) and therefore more representative of MSA evaluation. For EGY (ARZTB), we follow the recommended splits of Diab et al. (2013). For GLF (CamelTB-Gumar), we use our predefined splits (Section 4.4).

		#Sentences	#Tree Tokens
DEV	MSA	2,022	35,418
	EGY	3,542	57,360
	GLF	435	4,112
TEST	MSA	1,918	34,757
	EGY	3,718	57,995
	GLF	425	3,756

Table 4: **DEV** and **TEST** set statistics used in our evaluation. MSA refers to CamelTB, EGY to ARZTB, and GLF to CamelTB-Gumar. We report the number of sentences and tree tokens for each variant.

Training Data	#Sentences	#Tree Tokens
MSA _{PATB, CamelTB}	25,186	762,554
EGY _{ARZTB}	24,428	393,193
GLF _{CamelTB-Gumar}	2,021	17,648
MSA+EGY	49,614	1,155,747
MSA+GLF	27,207	780,202
EGY+GLF	26,449	410,841
MSA+EGY+GLF	51,635	1,173,395

Table 5: **TRAIN** data statistics for the MSA (PATB and CamelTB), EGY (ARZTB), and GLF (CamelTB-Gumar) treebanks, as well as their aggregated combinations. We report the total number of sentences and dependency tree tokens used in each setup.

Training Data and its Combination For training, we use PATB and CamelTB train splits as a base representative for MSA training. This matches the training setup of CamelParser2.0. For EGY (ARZTB), we follow the recommended training splits of Diab et al. (2013). For GLF (CamelTB-Gumar), we use our predefined training splits (Section 4.4). Furthermore, we examine how training data composition affects cross-variety parsing through controlled comparisons. We train single-variety models (MSA, EGY, GLF) as in-domain baselines; two-variety models (MSA+EGY, MSA+GLF, EGY+GLF) to assess mixed setups; and an all-variety model (MSA+EGY+GLF) to test joint training. Architecture and hyperparameters are fixed; only the data mix varies. Training sizes are shown in Table 5.

Parameter	Value	Parameter	Value
bert_pooling	mean	mlp_dropout	0.33
clip	5.0	mu	0.9
encoder_dropout	0.1	n_arc_mlp	500
epochs	10	n_bert_layers	4
eps	1e-08	n_rel_mlp	100
fix_len	20	nu	0.999
lr	5e-5	update_steps	4
lr_rate	10	warmup	0.1
min_freq	2	weight_decay	0
mix_dropout	0.0		

Table 6: Hyperparameters used to train our parsing models.

5.2. Dependency Parser

We adopt a biaffine dependency parser (Dozat and Manning, 2017), implemented in SuPar (Zhang et al., 2020), using BERT-based embeddings as input. As a baseline comparison point, we include CamelParser2.0 (Elshabrawy et al., 2023), which uses CAMEL-BERT-MSA (Inoue et al., 2021) as its contextual encoder backbone. In our models, we instead use CAMEL-BERT-MIX for contextual embeddings, as it is better suited for handling multiple Arabic variants, following the recommendations of Inoue et al. (2021).

For training, we use the hyperparameter settings of CamelParser2.0 (Elshabrawy et al., 2023), summarized in Table 6 for reproducibility. All models are trained and evaluated on a single NVIDIA V100 GPU, with a total computational cost of approximately 200 GPU minutes. Each parsing model has approximately 110M parameters.

5.3. Dialect Identification for Subset-Level Evaluation

Arabic is inherently diglossic (Ferguson, 1959), and even datasets labeled as a particular dialect often contain a mixture of text from different variants. To enable dialect-specific analysis of parsing results, we estimate the distribution of predicted dialect labels using the dialect identification system of Salameh et al. (2018) implemented in CAMEL Tools (Obeid et al., 2020). We use MODEL-26 with regional-level labels, achieving an F_1 score of 84.0 on the evaluation set reported by Salameh et al. (2018). We select the dialect label with the highest predicted probability among the three target variants: MSA, EGY, and GLF.

5.4. Evaluation Metric

We report Labeled Attachment Score (LAS), defined as the percentage of tokens for which both the predicted head (parent) and the dependency relation label are correct.

		$ \mathcal{S}_{ALL} $	$\%S_{MSA}$	$\%S_{EGY}$	$\%S_{GLF}$
Dev	MSA	2,022	63.4	2.6	34.0
	EGY	3,542	8.4	55.3	36.3
	GLF	435	0.5	3.0	96.6
Test	MSA	1,918	64.6	2.9	32.5
	EGY	3,718	10.1	55.6	34.3
	GLF	425	0.7	4.7	94.6

Table 7: Distribution of dialectal varieties in the development and test sets. For each evaluation subset (MSA, EGY, GLF), we report the total number of sentences ($|\mathcal{S}_{ALL}|$) and the percentage of sentences identified as MSA, EGY, and GLF ($\%S_{MSA}$, $\%S_{EGY}$, $\%S_{GLF}$).

6. Results

In this section, we present parsing results across MSA, Egyptian (EGY), and Gulf (GLF) Arabic under different training configurations. We begin with dialect identification statistics to motivate subset-level evaluation, then compare single- and multi-variety models on DEV and TEST, and conclude with a detailed comparison against CamelParser2.0.

6.1. Dialect Identification

Table 7 presents the proportion of sentences predicted as MSA, EGY, and GLF in our evaluation sets. EGY (ARZTB) is mostly predicted as EGY, with a notable portion labeled as GLF. GLF (CamelTB-Gumar) is almost entirely predicted as GLF, while MSA (CamelTB) is largely predicted as MSA. These trends suggest that the predicted labels generally align with the expected dialectal focus of each dataset, while highlighting some degree of variety and possibly genre mixing, especially in ARZTB. When considering all datasets combined, the predicted dialect distribution is nearly balanced among the three variants. Given this, we evaluate performance by predicted dialect rather than by treebank, using subsets of sentences identified as MSA, EGY, and GLF (denoted $\mathcal{S}_{MSA}$, $\mathcal{S}_{EGY}$, and $\mathcal{S}_{GLF}$, respectively). For example, $\mathcal{S}_{MSA}$ denotes the set of sentences in a given evaluation dataset predicted as MSA, while $\mathcal{S}_{EGY}$ and $\mathcal{S}_{GLF}$ denote those predicted as EGY and GLF.

6.2. Dependency Parsing

DEV Set Results Table 8 reports LAS scores on the DEV sets under different training configurations. Overall, the full-data model (MSA+EGY+GLF) achieves the highest macro-average performance (83.1 in $\mathcal{S}_{ALL}$), improving over CamelParser2.0 by 6.1 LAS points. It also yields the strongest dialect-specific macro-averages ($\mathcal{S}_{MSA}$: 88.7, $\mathcal{S}_{EGY}$: 83.7, $\mathcal{S}_{GLF}$: 83.9).

Training Data	MSA _{DEV}				EGY _{DEV}				GLF _{DEV}				AVG			
	$\mathcal{S}_{ALL}$	$\mathcal{S}_{MSA}$	$\mathcal{S}_{EGY}$	$\mathcal{S}_{GLF}$	$\mathcal{S}_{ALL}$	$\mathcal{S}_{MSA}$	$\mathcal{S}_{EGY}$	$\mathcal{S}_{GLF}$	$\mathcal{S}_{ALL}$	$\mathcal{S}_{MSA}$	$\mathcal{S}_{EGY}$	$\mathcal{S}_{GLF}$	$\mathcal{S}_{ALL}$	$\mathcal{S}_{MSA}$	$\mathcal{S}_{EGY}$	$\mathcal{S}_{GLF}$
MSA	86.9	86.8	87.4	87.1	72.1	80.7	70.0	72.8	71.5	77.8	78.6	71.2	76.8	81.8	78.7	77.0
EGY	80.0	80.0	81.6	79.7	83.3	84.1	82.4	85.3	65.6	88.9	76.4	65.1	76.3	84.3	80.1	76.7
GLF	65.7	64.7	69.8	68.8	58.0	57.4	57.2	60.1	71.2	83.3	70.7	71.1	65.0	68.5	65.9	66.7
MSA+EGY	86.8	86.7	87.5	87.4	84.1	84.8	83.3	85.7	67.3	94.4	74.3	66.9	79.4	88.6	81.7	80.0
MSA+GLF	87.0	86.8	87.4	87.4	73.5	80.9	71.7	74.1	79.5	88.9	87.1	79.1	80.0	85.5	82.1	80.2
EGY+GLF	80.7	80.6	82.9	80.8	83.3	84.3	82.3	85.2	76.9	88.9	77.1	76.8	80.3	84.6	80.8	80.9
MSA+EGY+GLF	86.9	86.7	87.4	87.2	83.9	85.0	82.9	86.0	78.6	94.4	80.7	78.4	83.1	88.7	83.7	83.9
CamelParser2.0	87.2	87.1	87.5	87.6	72.4	80.9	70.0	73.7	71.4	77.8	77.1	71.2	77.0	81.9	78.2	77.5

Table 8: **LAS** scores on the **DEV** sets for models trained on various data combinations evaluated on the MSA, EGY, and GLF. We report overall scores ($\mathcal{S}_{ALL}$) as well as dialect-specific subset scores ($\mathcal{S}_{MSA}$, $\mathcal{S}_{EGY}$, $\mathcal{S}_{GLF}$). The AVG column shows the macro-average across the three DEV sets. Bold indicates the best result within each column.

Training Data	MSA _{TEST}				EGY _{TEST}				GLF _{TEST}				AVG			
	$\mathcal{S}_{ALL}$	$\mathcal{S}_{MSA}$	$\mathcal{S}_{EGY}$	$\mathcal{S}_{GLF}$	$\mathcal{S}_{ALL}$	$\mathcal{S}_{MSA}$	$\mathcal{S}_{EGY}$	$\mathcal{S}_{GLF}$	$\mathcal{S}_{ALL}$	$\mathcal{S}_{MSA}$	$\mathcal{S}_{EGY}$	$\mathcal{S}_{GLF}$	$\mathcal{S}_{ALL}$	$\mathcal{S}_{MSA}$	$\mathcal{S}_{EGY}$	$\mathcal{S}_{GLF}$
MSA	87.3	87.0	90.6	88.2	73.0	81.4	70.7	73.6	73.3	66.7	72.5	73.3	77.9	78.4	77.9	78.4
EGY	79.2	79.5	86.3	77.4	83.9	85.3	83.1	84.8	68.7	75.0	65.6	68.8	77.3	79.9	78.3	77.0
GLF	65.4	64.9	71.7	66.6	58.7	58.4	58.4	59.5	73.8	66.7	77.3	73.7	66.0	63.3	69.1	66.6
MSA+EGY	87.1	86.9	91.2	87.5	84.4	85.7	83.6	85.3	70.1	66.7	68.3	70.2	80.5	79.8	81.0	81.0
MSA+GLF	87.2	86.9	90.6	88.2	74.4	81.3	72.7	74.0	81.0	83.3	79.9	81.1	80.9	83.8	81.1	81.1
EGY+GLF	80.0	80.0	86.5	79.3	83.8	85.0	83.1	84.8	79.4	75.0	73.0	79.7	81.1	80.0	80.9	81.3
MSA+EGY+GLF	87.2	86.9	91.8	87.7	84.2	86.1	83.6	84.6	80.3	66.7	78.3	80.5	83.9	79.9	84.6	84.3
CamelParser2.0	87.5	87.3	91.0	88.0	73.3	82.2	70.9	73.5	72.7	66.7	72.5	72.7	77.8	78.7	78.1	78.1

Table 9: **LAS** on the **TEST** sets for models trained on different data combinations. Bold indicates the best result within each column.

Examining individual DEV sets reveals three main patterns. **First**, single-variety training limits cross-variety generalization. The MSA-only model performs well on MSA_{DEV} (86.9) but drops markedly on EGY and GLF, and similar degradation is observed for EGY-only and GLF-only setups. Our MSA-only model is also slightly below CamelParser2.0 under the same setting, likely due to the pretrained encoder choice: as noted by Inoue et al. (2021), CAMELBERT-MSA (used by CamelParser2.0) is optimal for pure MSA, whereas CAMELBERT-MIX (used here) favors dialectal and mixed scenarios, prioritizing robustness over peak MSA performance. **Second**, combining varieties consistently improves robustness. In all cases, two-variety models outperform their single-variety counterparts on the relevant DEV sets (e.g., MSA+GLF over GLF-only on GLF_{DEV}), indicating positive cross-variety transfer rather than interference. **Third**, even small amounts of additional dialectal data can have a large impact. Adding GLF to MSA+EGY substantially boosts GLF performance (67.3→78.6 in $\mathcal{S}_{ALL}$) with minimal effect elsewhere, suggesting that limited low-resource data can effectively leverage transfer from higher-resource varieties. Although some two-variety models achieve

slightly higher scores on specific subsets, the differences are marginal. Overall, the full-data model offers the best macro-average and the most balanced cross-variety performance, supporting a single unified parser.

TEST Set Results Table 9 confirms that the trends observed on DEV generalize to TEST. The full-data model (MSA+EGY+GLF) achieves the highest overall macro-average (83.9 in $\mathcal{S}_{ALL}$), substantially outperforming CamelParser2.0 (77.8 AVG). It also yields the best dialect-specific averages ($\mathcal{S}_{EGY}$: 84.6, $\mathcal{S}_{GLF}$: 84.3). We note that $\mathcal{S}_{MSA}$ on GLF_{TEST} is much lower than on DEV; however, $\mathcal{S}_{MSA}$ sentences constitute less than 1% of the GLF subsets (Table 7), making this metric highly sensitive to small fluctuations and not indicative of overall trends. As in DEV, single-variety models perform best in-domain but degrade elsewhere, whereas multi-variety training improves cross-variety robustness. The unified model achieves the best overall average and remains competitive with the dedicated MSA baseline on MSA_{TEST}, reflecting the same trade-off observed on DEV. Overall, TEST results confirm that incor-

Section	Metric	MSA _{TEST}		EGY _{TEST}		GLF _{TEST}		Avg △
		BEST	CP2.0	BEST	CP2.0	BEST	CP2.0	
Overall	UAS	89.6	89.8	86.7	78.4	82.5	76.8	+4.6
	LS	92.9	93.1	91.6	83.1	85.1	80.6	+4.3
	LAS	87.2	87.5	84.2	73.2	80.3	72.7	+6.1
Relation	MOD	94.7	94.9	93.2	85.6	85.6	80.7	+4.1
	—	79.3	79.9	80.1	61.5	74.1	72.1	+6.7
	OBJ	94.1	94.3	92.8	86.4	90.3	86.6	+3.3
	IDF	97.1	97.1	95.2	91.8	97.1	91.3	+3.1
	SBJ	89.2	89.1	87.6	77.7	83.1	77.5	+5.1
	PRD	87.9	87.8	89.2	74.4	88.6	70.9	+10.9
	TPC	35.6	39.1	68.2	23.8	55.2	38.1	+19.3
	TMZ	69.3	64.2	33.3	6.1	—	—	+16.2
Root	F-score	77.3	77.4	80.4	71.4	68.1	66.1	+3.6
	Precision	93.1	93.2	88.1	78.2	95.5	92.7	+4.2
	Recall	66.0	66.1	73.9	65.6	52.9	51.3	+3.2

Table 10: Comparison between our multi-variety **BEST** system setup (MSA+EGY+GLF) and CamelParser2.0 (**CP2.0**) on MSA, EGY, and GLF test sets. Results are organized by overall accuracy, relation-level F-scores, and ROOT identification. Cells marked “—” indicate that the relation did not occur in the dataset. Bold indicates higher scores; △ denotes the average improvement of BEST over CP2.0.

porating all dialectal data yields the most balanced and strongest parser across varieties.

6.3. Analysis

To help us understand where the improvement in performance is happening, we compare our **BEST** system setup (MSA+EGY+GLF) and CamelParser2.0 across MSA, EGY, and GLF Test sets (Table 9). We present our analysis from three perspectives: (1) overall parsing accuracy in terms of Unlabeled Attachment Score (UAS) and Label Score (LS) in addition to LAS, (2) relation-level F-score, and (3) ROOT parent identification. While CamelParser2.0 remains highly competitive on MSA, BEST yields substantial gains on dialectal Arabic without sacrificing MSA performance. The detailed results are presented in Table 10.

Overall Accuracy BEST slightly trails CamelParser2.0 on MSA in UAS (89.6 vs. 89.8) but yields substantial gains on dialectal data: +8.3 on EGY and +5.7 on GLF. A similar pattern holds for LS, with near parity in MSA (92.9 vs. 93.1) and clear improvements in EGY (+8.5) and GLF (+4.5). These gains culminate in LAS, where BEST improves markedly on EGY (+11.0) and GLF (+7.6) while remaining comparable on MSA (87.2 vs. 87.5).

Relation-Level Analysis BEST consistently outperforms CamelParser2.0 on both dialectal varieties, while remaining largely on par in MSA. The largest gains occur in syntactically marked constructions. Predicate (PRD), Topic (TPC), and Tamyiz (TMZ), the three least frequent relations,

show the greatest relative improvements: +10.9, +19.3, and +16.2, respectively. Core grammatical relations (SBJ, OBJ, IDF) also exhibit steady gains (3–5 points), indicating broad and systematic improvements rather than isolated effects.

Root Identification ROOT parent identification shows clear gains in EGY and modest improvements in GLF, while remaining on par in MSA. In EGY, F-score increases by +9.0, driven by improvements in both precision (+9.9) and recall (+8.3), reflecting more accurate clause boundary detection and predicate selection. In GLF, gains are smaller (+2.0 F_1), with very high precision (95.5) but considerably lower recall (52.9). This precision–recall imbalance is consistent with the previously discussed fragmentation observed in CamelTB-Gumar sentences (Section 4.4).

Overall, our BEST setup outperforms CamelParser2.0 on average across test sets, driven by substantial improvements on EGY and GLF, while remaining only marginally lower on MSA. These results demonstrate robust cross-variety improvements without materially sacrificing performance on standard Arabic. Nevertheless, there remains room for improvement across all varieties.

7. Conclusion and Future Work

This work revisits dialectal Arabic parsing, a task that has seen limited progress despite the well-documented shortcomings of MSA-trained parsers on dialectal data. We present new empirical results on Egyptian and Gulf Arabic, demonstrating that

even modest amounts of dialectal annotation yield substantial improvements in parsing accuracy. Our contributions include a newly annotated Gulf Arabic dataset, a strong multi-variety Arabic parser, and a dialect identification–based analytical framework that clarifies how training data influences performance across dialects and evaluation settings.

In the future, we aim to move toward a truly pan-Arabic parser capable of robust generalization across varieties. This will involve expanding annotation to additional dialects and genres, evaluating full pipelines from raw text, and further examining cross-variety transfer effects. We also plan to use the new treebanks to systematically assess large language models on Arabic parsing. More broadly, by releasing datasets, models, and tools, we hope to foster a more inclusive and empirically grounded ecosystem for Arabic NLP research.

Acknowledgments

We thank the anonymous reviewers for their insightful and constructive comments. We also acknowledge the support of the High Performance Computing Center at New York University Abu Dhabi.

Limitations

Our work is limited by the availability of annotated data for Arabic dialects. Although we introduce a new evaluation set for Gulf Arabic, it remains relatively small compared to MSA treebanks. Parsing performance is also bounded by the noise and variability in dialectal orthography, especially in user-generated content. Additionally, while dialect identification helps analyze parsing behavior, it may introduce analysis errors in cases of code-switching or mixed dialect input. Finally, our models have been trained and evaluated on relatively clean, segmented input, and their robustness to noisy or unsegmented text is left to future work.

Ethical Considerations

Working with dialectal Arabic raises several ethical considerations. First, dialects are regionally and socially marked; care must be taken to avoid reinforcing stigmas or privileging certain varieties over others. Our choice to include Gulf and Egyptian Arabic was driven by data availability, but we acknowledge that this leaves out many underrepresented dialects. Second, our Gulf Arabic dataset was annotated by Arabic native speakers, and we ensured they were fairly compensated for their work. We emphasize the importance of fair payment when working with community annotators on dialectal resources. Finally, as dialectal tools become more widespread, they may be used for surveillance or social media

monitoring. We urge future users and developers to consider these implications and adopt responsible use policies.

8. Bibliographical References

- Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. [QADI: Arabic dialect identification in the wild](#). In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.
- Muhammad Abdul-Mageed, Amr Keleg, Abdel-Rahim Elmadany, Chiyu Zhang, Injy Hamed, Walid Magdy, Houda Bouamor, and Nizar Habash. 2024. [NADI 2024: The fifth nuanced Arabic dialect identification shared task](#). In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 709–728, Bangkok, Thailand. Association for Computational Linguistics.
- Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. [NADI 2021: The second nuanced Arabic dialect identification shared task](#). In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.
- Noor Abo Mokh, Daniel Dakota, and Sandra Kübler. 2024. [Out-of-domain dependency parsing for dialects of Arabic: A case study](#). In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 170–182, Bangkok, Thailand. Association for Computational Linguistics.
- Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. [Shami: A corpus of Levantine Arabic dialects](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Sharefah Al-Ghamdi, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2021. [A dependency treebank for classical Arabic poetry](#). In *Proceedings of the Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)*, pages 1–9, Sofia, Bulgaria.
- Sharefah Al-Ghamdi, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2023. [Fine-tuning BERT-based pre-trained models for Arabic dependency parsing](#). *Applied Sciences*, 13(7).

- Israa Alsarsour, Esraa Mohamed, Reem Suwaileh, and Tamer Elsayed. 2018. DART: A large dataset of dialectal Arabic tweets. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.
- Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2022. [Hierarchical aggregation of dialectal data for Arabic dialect identification](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4586–4596, Marseille, France. European Language Resources Association.
- Houda Bouamor, Hanan Alshikhabobakr, Behrang Mohit, and Kemal Oflazer. 2014. A Human Judgment Corpus and a Metric for Arabic MT Evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 207–213, Doha, Qatar.
- Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR Shared Task on Arabic Fine-Grained Dialect Identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19)*, Florence, Italy.
- David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. [Parsing Arabic dialects](#). In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 369–376, Trento, Italy. Association for Computational Linguistics.
- Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavall-Sforza, Samhaa R El-Beltagy, Wassim El-Hajj, et al. 2021. A panoramic survey of natural language processing in the Arab world. *Communications of the ACM*, 64(4):72–81.
- Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. [LDC Arabic treebanks and associated corpora: Data divisions manual](#).
- Timothy Dozat and Christopher D. Manning. 2017. [Deep biaffine attention for neural dependency parsing](#).
- Kais Dukes and Tim Buckwalter. 2010. A Dependency Treebank of the Quran using Traditional Arabic Grammar. In *Proceedings of the Conference on Informatics and Systems (INFOS)*, Cairo, Egypt.
- Ahmed Elshabrawy, Muhammed AbuOdeh, Go Inoue, and Nizar Habash. 2023. [CamelParser2.0: A state-of-the-art dependency parser for Arabic](#). In *Proceedings of ArabicNLP 2023*, pages 170–180, Singapore (Hybrid). Association for Computational Linguistics.
- Asma Etman and Louis Beex. 2015. Language and Dialect Identification: A Survey. In *Proceedings of the Intelligent Systems Conference (IntelliSys)*, London, UK.
- Rui Fan, Shu Li, Tingting He, and Yu Liu. 2025. [Aspect-based sentiment analysis with syntax-opinion-sentiment reasoning chain](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3123–3137, Abu Dhabi, UAE. Association for Computational Linguistics.
- Charles F Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.
- Ping Guo, Yubing Ren, Yue Hu, Yunpeng Li, Jiarui Zhang, Xingsheng Zhang, and Heyan Huang. 2024. [Teaching large language models to translate on low-resource languages with textbook prompting](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15685–15697, Torino, Italia. ELRA and ICCL.
- Nizar Habash, Muhammed AbuOdeh, Dima Taji, Reem Faraj, Jamila El Gizuli, and Omar Kallas. 2022. [Camel treebank: An open multi-genre Arabic dependency treebank](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2672–2681, Marseille, France. European Language Resources Association.
- Nizar Habash, Reem Faraj, and Ryan Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.
- Nizar Habash and Ryan Roth. 2009. [CATiB: The Columbia Arabic treebank](#). In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore. Association for Computational Linguistics.
- Dana Halabi, Ebaa Fayyoubi, and Arafat Awajan. 2021. I3rab: A new Arabic dependency treebank based on Arabic grammatical theory. *Transactions on Asian and Low-Resource Language Information Processing*, 21(2):1–32.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. [The interplay of variant, size, and task type in Arabic pre-trained language models](#). In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

- Hiroshi Kanayama, Yang Zhao, Ran Iwamoto, and Takuya Ohko. 2024. [Incorporating syntax and lexical knowledge to multilingual sentiment classification on large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4810–4817, Bangkok, Thailand. Association for Computational Linguistics.
- Yash Kankanampati, Joseph Le Roux, Nadi Tomeh, Dima Taji, and Nizar Habash. 2020. [Multitask easy-first dependency parsing: Exploiting complementarities of different dependency representations](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2497–2508, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Salam Khalifa, Nizar Habash, Dana Abdulrahim, and Sara Hassan. 2016. [A large scale corpus of Gulf Arabic](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 4282–4289, Portorož, Slovenia. European Language Resources Association (ELRA).
- Salam Khalifa, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi. 2018. A morphologically annotated corpus of Emirati Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.
- Xiang Li, Zhiyi Yin, Hexiang Tan, Shaoling Jing, Du Su, Yi Cheng, Huawei Shen, and Fei Sun. 2025. [PRDetect: Perturbation-robust LLM-generated text detection based on syntax tree](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8290–8301, Albuquerque, New Mexico. Association for Computational Linguistics.
- Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. 2006. Developing and Using a Pilot Dialectal Arabic Treebank. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Genoa, Italy.
- Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.
- Mohamed Maamouri, Ann Bies, Seth Kulick, Michael Ciul, Nizar Habash, and Ramy Eskander. 2014. [Developing an Egyptian Arabic Treebank: Impact of dialectal morphology on annotation and tool development](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).
- Yuval Marton, Nizar Habash, and Owen Rambow. 2013. [Dependency parsing of Modern Standard Arabic with lexical and inflectional features](#). *Computational Linguistics*, 39(1):161–194.
- Benjamin Muller, Benoit Sagot, and Djamé Seddah. 2020. [Can multilingual language models transfer to an unseen dialect? a case study on North African Arabizi](#).
- Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 2216–2219, Genoa, Italy.
- Ossama Obeid, Go Inoue, and Nizar Habash. 2022. [Camelira: An Arabic multi-dialect morphological disambiguator](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–326, Abu Dhabi, UAE. Association for Computational Linguistics.
- Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. [CAMEL tools: An open source python toolkit for Arabic natural language processing](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.
- Ahmed Salama, Houda Bouamor, Behrang Mohit, and Kemal Oflazer. 2014. YouDACC: the Youtube Dialectal Arabic Comment Corpus. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 1246–1251, Reykjavik, Iceland.
- Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. [Fine-grained Arabic dialect identification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab. 2014. [Sentence level dialect identification for machine translation system selection](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 772–778, Baltimore, Maryland. Association for Computational Linguistics.

- Djamé Seddah, Farah Essaidi, Amal Fethi, Matthieu Futral, Benjamin Muller, Pedro Javier Ortiz Suárez, Benoît Sagot, and Abhishek Srivastava. 2020. [Building a user-generated content North-African Arabizi treebank: Tackling hell](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1139–1150, Online. Association for Computational Linguistics.
- Anas Shahrour, Salam Khalifa, Dima Taji, and Nizar Habash. 2016. [CamelParser: A system for Arabic syntactic analysis and morphological disambiguation](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 228–232, Osaka, Japan. The COLING 2016 Organizing Committee.
- Otakar Smrž, Jan Šnaidauf, and Petr Zemánek. 2002. Prague Dependency Treebank for Arabic: Multi-Level Annotation of Arabic Corpus. In *Proceedings of the International Symposium on Processing of Arabic*, pages 147–155, Manouba, Tunisia.
- Dima Taji, Nizar Habash, and Daniel Zeman. 2017. Universal dependencies for Arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, Valencia, Spain.
- Wajdi Zaghouani and Anis Charfi. 2018. ArapTweet: A Large Multi-Dialect Twitter Corpus for Gender, Age and Language Variety Identification. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.
- Omar F Zaidan and Chris Callison-Burch. 2011. The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic With High Dialectal Content. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 37–41.
- Yu Zhang, Zhenghua Li, and Min Zhang. 2020. [Efficient second-order TreeCRF for neural dependency parsing](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online. Association for Computational Linguistics.

On LLM Prompting Techniques for Arabic Language Arithmetic Reasoning

Reem Khaled Alenezi, Ayed Atallah Salman

Department of Computer Engineering, Kuwait University
Kuwait City, Kuwait
r.alenezi@grad.ku.edu.kw, ayed.salman@ku.edu.kw

Abstract

Math word problems (MWP) require complex reasoning to extract mathematical relationships from textual descriptions. While Large Language Models (LLMs) have shown remarkable performance on English mathematical reasoning tasks, their effectiveness on Arabic MWPs remains largely unexplored. This paper introduces three Arabic datasets (AGSM8K, Qudurat, and ArabicMWPs) and evaluates six LLMs using three prompting techniques: Manual Chain-of-Thought (CoT), Zero-shot CoT, and Self-consistency. Performance is assessed using accuracy and BERTScore metrics (precision, recall, F1-score). Our findings demonstrate that GPT-4o with Self-consistency achieves the highest accuracy of 97.65% on AGSM8K. It also obtains a precision of 71.94%, a recall of 71.31%, and an F1-score of 71.50%. The Arabic-specific LLM ALLaM achieves 84.41% accuracy on ArabicMWPs and 43.97% on AGSM8K. Fine-tuning experiments are further conducted on models using Arabic mathematical data. This work addresses the critical gap in Arabic mathematical reasoning resources and provides insights for developing Arabic-capable AI systems. Prompt-engineering methods combined with LLMs are regarded as a strong approach for advancing education and scientific research in solving Arabic mathematical problems.

Keywords: Large Language Models, Prompt Engineering, Mathematical Reasoning, Chain-of-Thought, Self-Consistency, Arabic Language

1. Introduction

Math Word Problems (MWPs) require integrating language comprehension with mathematical reasoning, making them critical for mathematical education. Large Language Models (LLMs) have demonstrated significant progress in various natural language processing (NLP) tasks. MWPs are questions described through a written context, and the solution is found by using mathematical equations. Solving MWPs requires accurate comprehension, analytical reasoning, and information extraction (Lu et al., 2022). MWPs are used in education, scientific research and everyday life.

Prompt engineering plays an important role in improving LLM performance by designing effective prompts that guide models toward optimal solutions, enabling them to handle complex tasks more effectively without additional training. (Amatriain, 2024). The Chain-of-Thought (CoT) technique, which mimics human logical reasoning by breaking down a problem into steps, has been employed to solve this type of task. CoT prompting can be implemented as a few-shot examples or zero-shot (Wei et al., 2022; Kojima et al., 2022). Additionally, for more complex mathematical tasks, the self-consistency prompting method generates multiple potential solutions and selects the most consistent response (Wang et al., 2022).

Arabic, spoken by over 300 million native speakers (World Population Review, 2024), presents unique linguistic challenges for MWP solving due to its rich morphology, dialectal

variations, and frequent absence of diacritics. Despite progress in MWP solving using prompting techniques, most advancements focus on non-Arabic datasets (Ahn et al., 2024). Arabic MWP resources remain scarce, and advanced prompting methods have not been systematically explored for Arabic mathematical reasoning tasks.

This study addresses a critical gap in Arabic NLP by evaluating LLM capabilities in solving Arabic MWPs using three prompting techniques (Manual CoT, Zero-shot CoT, Self-Consistency). The research examines how different LLMs understand Arabic mathematical problems, extract numerical relationships, and generate coherent reasoning steps. The primary objective is to compare the performance of closed-source and open-source LLMs using different prompting methods to identify the most effective approach for Arabic mathematical reasoning.

Our contributions include: (1) Introduction of three new Arabic MWP with varying difficulty levels and question types, (2) Comprehensive evaluation of six LLMs using three prompting techniques on Arabic mathematical reasoning, (3) Analysis of fine-tuning effects on Arabic mathematical reasoning performance, and (4) Identification of optimal configurations for Arabic MWP solving.

To provide the necessary background, the next section reviews key prompting techniques before introducing the methodology and experiments.

2. Prompt Engineering

Prompt engineering is the process of designing prompts that guide LLMs to correctly understand the task and produce accurate responses. After creating an effective prompt, it is passed to the LLM to obtain a better response (Liu et al., 2023). This technique allows researchers to use the prompt as an input with modifications to perform various tasks, which is easier than training the model from scratch for each task.

2.1 Standard Prompting

Studies have shown that LLM outputs are highly influenced by prompt design, highlighting the importance of crafting prompts that align with task requirements. A prompt may simply consist of instructions or a direct question. Zero-shot prompting relies solely on a natural language task description without examples, which is effective for simple tasks but often insufficient for problems requiring multi-step reasoning (Brown et al., 2020; Liu et al., 2023). A Few-shot prompting technique improves performance by incorporating a small number of illustrative examples that guide the model's reasoning and output structure. When this approach remains insufficient for complex mathematical problems, tasks can be decomposed into intermediate steps using Chain-of-Thought (CoT) prompting.

2.2 Chain-of-thought Prompting

Chain-of-Thought prompt method was proposed by Wei et al. (2022) to improve LLMs' ability to understand complex reasoning NLP task. This is achieved by providing the model with a prompt to generate intermediate reasoning steps to think about the question logically before giving the final answer. The main idea of a CoT prompt is to emulate human thinking when attempting to solve a complex problem. An effective approach is to break the question into smaller parts and solve each component individually. This is done by thinking step by step in a logical way before getting the outcome. In few-shot CoT, the model is provided with question-and-answer examples with intermediate logical steps. The CoT method depends on manually constructed examples, leading to its alternative name, Manual CoT (Wei et al., 2022).

Another advanced prompting method used for reasoning tasks is Zero-shot CoT (Kojima et al., 2022). Unlike manual CoT, this technique does not provide explicit reasoning examples. Instead, it relies on adding the phrase "Let's think step by step" to the prompt. This simple instruction has been shown to encourage the LLM to generate intermediate reasoning steps before producing the final answer, thereby improving its logical performance.

2.3 Self-Consistency Prompting

Self-consistency is an advanced prompting technique that is used to enhance the LLMs abilities when dealing with logical and reasoning tasks (Wang et al., 2022). LLMs generate a single answer to the question when using traditional prompting techniques like CoT. These techniques may cause logical errors, especially with complicated tasks. To avoid this issue, self-consistency, which contains two main ideas: sampling and majority voting, is applied to LLM. This method works by generating multiple reasoning paths with different answers for the same problem, then selecting the most frequent answer as the final output. The number of reasoning paths affects the model's performance. Figure 1 outlines self-consistency method.

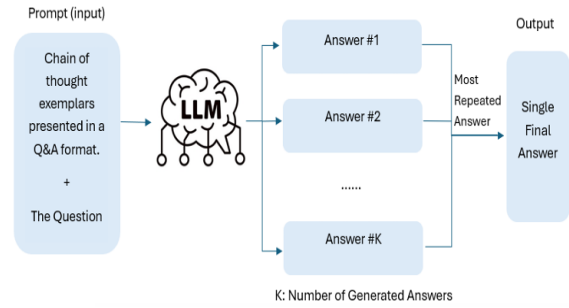


Figure 1: Self-consistency prompting method

The next section discusses prior studies related to Arabic language tasks and the use of prompting techniques for mathematical reasoning.

3. Related Work

3.1 Mathematical Reasoning with LLMs

There have been various research studies that have proposed prompting techniques for solving MWPs. These methods have produced different results in terms of accuracy and the responsiveness of the LLM when solving problems. The Chain-of-Thought (CoT) technique has shown impressive results when used with LLMs to solve MWPs that require logical reasoning to find the solution. Few-shot CoT provides the model with manually created demonstrations to generate intermediate steps to think logically about the question (Wei et al., 2022).

Kojima et al. introduced Zero-shot CoT, showing that simply adding "Let's think step by step" can elicit reasoning without examples. Wang et al. proposed Self-consistency, which samples multiple reasoning paths and selects the most frequent answer through majority voting. This approach achieved substantial improvements over standard CoT on various reasoning benchmarks.

Another prompt technique is presented called Plan-and-Solve prompting (Wang et al., 2023). In this method, LLM generates a plan to solve a problem then the model applies this plan to find the answer for the question correctly step by step. Recent research includes Active Prompting (Diao et al., 2023), which adaptively selects examples based on uncertainty, and Tree of Thoughts (Yao et al., 2023), which explores multiple reasoning branches systematically.

These previous studies highlight different prompting techniques that have been used to solve MWPs and have demonstrated significant improvements in the performance of LLMs.

Most previous studies on solving mathematical word problems using prompting techniques have been conducted in non-Arabic languages, largely due to the lack of widely available Arabic benchmark datasets. Consequently, research on Arabic mathematical word problems remains limited despite the importance of the field and the growing role of artificial intelligence in mathematical reasoning tasks. Recent work has also introduced Arabic-specific mathematical reasoning datasets, such as ArMATH (Alghamdi et al., 2022), which provides a benchmark for solving Arabic mathematical word problems using deep learning techniques.

One of the Arabic MWPs datasets is QIYAS which mainly contain multiple-choice math problems that are typically solved using zero, one, and few-shot prompting (Al-Khalifa & Al-Khalifa, 2024). In contrast, this work examines not only the final answer but also the generated reasoning steps. To the best of our knowledge, no study has systematically evaluated multiple prompting strategies across both closed-source and open-source LLMs on Arabic MWP datasets. This study addresses this gap by benchmarking performance across three Arabic datasets with varying levels of difficulty.

To place these prompting techniques within the Arabic context, the following subsection reviews relevant research in Arabic NLP.

3.2 Arabic Natural Language Processing

Artificial intelligence has made significant progress in NLP, with modern language models demonstrating multilingual comprehension and high accuracy across diverse tasks. Arabic requires special attention due to its complex morphology and distinctive linguistic features, prompting increased research focus on developing AI-based tools for Arabic text processing, including sentiment analysis and machine translation (Zouidine & Khalil, 2025), named entity recognition (Albahli, 2025), question answering (Basem et al., 2025), and grammatical

error correction (Kwon et al., 2023), and text generation (Allam et al., 2025).

4. Methodology

4.1 Datasets

Three Arabic mathematical reasoning datasets are introduced in this work with different levels of difficulty:

AGSM8K: The Arabic Grade School Math 8K (AGSM8K) dataset is a novel dataset translated from the English GSM8K dataset (Cobbe et al., 2021). AGSM8K in Modern Standard Arabic consists of 7,473 training questions and 1,319 test questions, requiring between two and eight steps to solve.

Questions were translated using a ChatGPT model by providing it with a specific prompt designed to ensure that the final translation output followed the JSONL format. Manual review of all translations ensured mathematical consistency, eliminated translation errors, and validated that question meaning remained intact. After completing the translation and initial review, the translated questions were given to a language reviewer to ensure the correctness of the translation.

Qudurat: A novel dataset of 140 Academic Aptitude Test problems from Kuwait University (Kuwait University, 2024). The number of steps in Qudurat questions ranges between 1 and 8 to find the final answer. Some resources of these questions include details of how to solve with logical steps. There were some questions with a final answer without any steps, so a specialist in mathematics was assigned to solve them in detail. To find solutions to these mathematic problems, basic arithmetic operations (addition, subtraction, multiplication, and division) are used. Additionally, one question involved comparison using “less than” and “greater than” relations.

ArabicMWPs: A novel dataset contains 725 MWPs in Arabic. It is translated from two English MWPs datasets: AddSub (Hosseini et al., 2014) and MultiArith (Roy & Roth, 2016). AddSub is a simple dataset that needs only addition and subtraction to solve questions. While MultiArith contains questions need more calculations. AddSub has 395 questions and MultiArith consists of 600 questions. ArabicMWPs include 725 questions, 265 from AddSub and 460 from MultiArith.

These datasets can support educational applications by enabling students to practice and develop new problem-solving strategies for Arabic MWPs. They can also assist educators in teaching mathematical reasoning.

4.2 Models and Prompting Techniques

Six LLMs were evaluated in the main experiments: four closed-source (GPT-4o, GPT-

3.5-turbo, GPT-4o-mini, O3-mini) and two open-source (Llama-3.1-8B-Instruct, Qwen2.5-Coder-7B-Instruct) models. In addition to Arabic-Specific Model (ALLaM) (Bari et al., 2024).

Three prompting techniques were employed:

1. **Manual CoT:** Eight manually few-shot examples from (Wei et al., 2022) for English, while translated these examples for Arabic.
2. **Zero-shot CoT:** Adding (Let's think step by step) for English and ("دعونا نفكر خطوة بخطوة") for Arabic.
3. **Self-consistency:** Generating multiple reasoning paths ($K=10, 20, 40$) for Arabic and selecting majority answer. For English ($K=10$ and 20).

4.3 Experimental Setup

All experiments were conducted using consistent settings across models. Temperature values of 0.0 and 0.5 were applied for closed-source models, while 0.1 and 0.5 were used for open-source models. Experiments were executed on Google Colaboratory using an NVIDIA A100 GPU with a Jupyter Notebook environment, and all code was implemented in Python.

For self-consistency prompting, multiple reasoning paths were sampled using fixed values of K (e.g., 10, 20, and 40), and the final answer was determined through majority voting. To ensure reproducibility, a single self-consistency prompting template was used across all experiments, with identical temperature settings, while only the underlying language model was varied.

4.4 Evaluation Approach

In the main experiments, AGSM8K is used for Arabic MWPs and GSM8K for English MWPs. Performance of LLMs is evaluated using accuracy, precision, recall, and F1-score metrics.

Accuracy measures the LLM's ability to answer correctly by calculating the number of correct questions divided by the total number of questions.

BERTScore is used to evaluate the quality of the generated output. It reports precision, recall, and F1-score (Zhang et al., 2019):

- Precision: the average similarity between each token in the candidate text and the most similar token in the reference.
- Recall: the average similarity between each token in the reference text and the most similar token in the candidate.

- F1-score: the harmonic mean of precision and recall.

Not all temperature values and experimental settings are presented; only the most informative configurations are presented to maintain clarity and highlight meaningful trends.

4.5 Fine-tuning

Fine-tuning experiments were conducted on the AGSM8K dataset using a 90% training and 10% validation split, with appropriate hyperparameters applied for each model type. Fine-tuning of the three closed-source models (GPT-3.5-turbo, GPT-4o, and GPT-4o-mini) were performed using OpenAI's fine-tuning API. In addition, two open-source models (Llama-3.1-8B and Qwen-3-8B) were fine-tuned using AdaLoRA (Zhang et al., 2023) with 4-bit NF4 quantization. Fine-tuning for the closed-source language models was relatively quick, with the slowest model being GPT-3.5-turbo, which completed training in approximately 3 hours. In contrast, fine-tuning the open-source language models requires significantly more time. Fine-tuning was not available for O3-mini.

4.5.1 Configuration for OpenAI Models

OpenAI models were fine-tuned using a batch size of 20 and trained for 16 epochs. A learning rate multiplier of $1e-3$ was applied to enable gradual adaptation to Arabic mathematical patterns. A seed value of 42 was used to ensure reproducibility across all fine-tuning experiments.

4.5.2 Configuration for Open-source Models

Open-source models were fine-tuned with a batch size of 2 and trained for 50 epochs to ensure adequate adaptation to Arabic mathematical patterns. A learning rate of $3e-4$ and weight decay of 0.001 were applied to balance effective learning with regularization against overfitting.

To support reproducibility for this work, all resources are publicly available at: <https://github.com/ReemAlenezi/Arabic-MWPs-Reasoning>

5. Main Results

5.1 Arabic AGSM8K Results

Table 1 shows the accuracy results across different prompting techniques on Arabic AGSM8K datasets. Self-consistency consistently outperformed other methods, with GPT-4o achieving 97.65% accuracy using Self-consistency with $K=40$ with precision 71.94%, recall 71.31%, and F1-score 71.50%. In addition, performance of ALLaM-7B is better than open-source models. Temperature is set to 0 for OpenAI models and 0.1 for open-source models.

Model	Manual CoT	Zero-shot	Self-consistency K=40
GPT-3.5-turbo	52.91	58.33	70.58
GPT-4o	94.47	95.07	97.65
GPT-4o-mini	79.53	86.50	91.36
O3-mini	93.94	94.85	96.13
Llama-3.1-8B	36.09	35.48	56.06
Qwen2.5-Coder	18.27	21.91	26.35
ALLaM-7B	42.38	38.52	--

Table 1: Accuracy (%) of LLMs across prompting methods on AGSM8K

5.2 Self-Consistency with Varying Sample Sizes

Table 2 examines the effect of increasing sample size (K) in Self-consistency prompting on AGSM8K. Increasing K generally improves accuracy, with larger gains observed for weaker models; however, performance tends to saturate or slightly degrade at higher sample sizes for stronger models. Temperature is set to 0 for OpenAI models and 0.1 for open-source models.

Model	Sc (K=10)	Sc (K=20)	Sc (K=40)
GPT-3.5-turbo	58.38	60.96	70.58
GPT-4o	95.22	96.29	97.65
GPT-4o-mini	88.86	90.60	91.36
O3-mini	95.15	95.76	96.13
Llama-3.1-8B	50.95	54.55	56.06
Qwen2.5-Coder	25.15	26.06	26.35

Table 2: Accuracy (%) of Self-consistency with different K values on AGSM8K

5.3 English GSM8K Results

Table 3 reports the accuracy of LLMs across different prompting methods on the English GSM8K dataset. Temperature is set to 0 for OpenAI models and 0.1 for open-source models.

Model	Manual CoT	Zero-shot	Self-consistency K=10
GPT-3.5-turbo	77.56	75.44	79.45
GPT-4o	93.78	93.40	96.33
GPT-4o-mini	87.95	82.41	91.13
O3-mini	98.48	97.27	98.48
Llama-3.1-8B	69.60	58.61	73.44
Qwen2.5-Coder	65.81	77.63	79.83

Table 3: Accuracy (%) of LLMs across prompting methods on GSM8K

Overall, self-consistency prompting with K=10 achieves the highest accuracy for most models. Closed-source models, particularly GPT-4o and O3-mini, demonstrate consistently strong performance across all prompting strategies. Notably, O3-mini achieves the highest accuracy at 98.48%. GPT-4o-mini also shows noticeable improvement under self-consistency compared to Zero-shot and Manual CoT prompting.

Open-source models achieve lower overall accuracy but benefit from self-consistency, which yields higher performance compared to other prompting methods. For cross-lingual comparison, self-consistency results at K=10 are reported consistently for both Arabic and English datasets.

5.4 Fine-tuning Results

Fine-tuning demonstrates a clear differential impact across model families. As shown in Table 4, open-source models benefit substantially from fine-tuning, with Qwen3-8B achieving the largest absolute improvement (+14.02%), followed by Llama 3.1-8B (+9.47%). These gains indicate that targeted training on Arabic mathematical data effectively enhances both language understanding and numerical reasoning in open-source models.

In contrast, closed-source models exhibit more modest changes after fine-tuning. GPT-3.5-turbo and GPT-4o-mini show moderate improvements, while GPT-4o maintains nearly identical performance before and after fine-tuning, suggesting that its pre-training already captures the necessary reasoning patterns for Arabic MWPs.

Model	Base Accuracy	Fine-tuned Accuracy	Change
GPT-3.5-turbo	58.33	63.43	+5.10
GPT-4o-mini	86.50	89.16	+2.66
GPT-4o	95.50	95.07	-0.43
Llama 3.1-8B	35.48	44.95	+9.47
Qwen3-8B	37.91	51.93	+14.02

Table 4: Accuracy (%) before and after fine-tuning on AGSM8K

5.5 Cross-Lingual Results Overview

Figure 2 provides a visual summary of model accuracy across Arabic (AGSM8K) and English (GSM8K) under various prompting techniques. The x-axis represents the evaluated models under different prompting techniques, while the y-axis shows accuracy scores. The prompting strategies include CoT, zero-shot CoT, and Self-consistency (K = 10) for both Arabic and English datasets.

LLMs demonstrate strong performance on mathematical word problems written in English (GSM8K). Open-source language models perform substantially better in English, achieving accuracies of approximately 52–83%, while their performance in Arabic ranges from about 17–54%.

Closed-source models demonstrate more balanced performance across both languages. Models such as GPT-4 and GPT-4o-mini achieve strong results on Arabic AGSM8K in certain settings, particularly when using Chain-of-Thought or Zero-shot CoT prompting. Additionally, O3-mini and GPT-3.5-turbo show consistently high and stable performance on English GSM8K across all prompting methods.

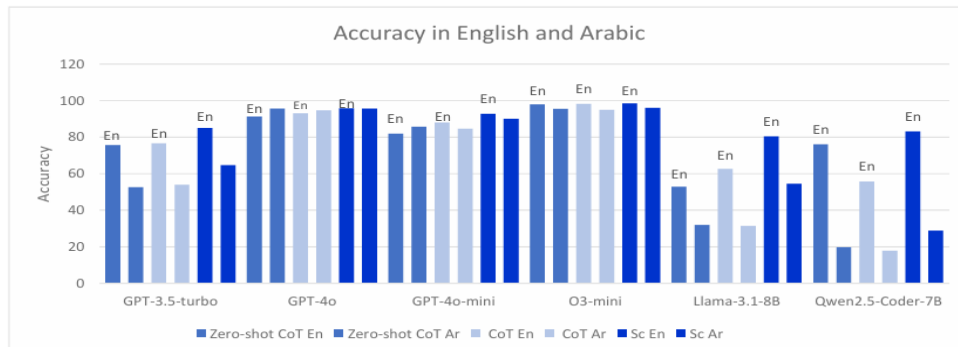


Figure 2: Accuracy of GSM8K and AGSM8K with LLMs

6. Discussion

6.1 Prompting Technique Effectiveness

The results demonstrated that the self-consistency method contributes to improving the performance of LLMs and are considered one of the most effective techniques for solving mathematical problems. This improvement is attributed to its underlying mechanism, which is based on generating multiple reasoning paths and selecting the most frequently occurring answer, thereby reducing the likelihood of errors caused by relying on a single reasoning path. Performance gains increase as the sample size grows, and this effect is particularly evident in language models with limited capabilities. This suggests that aggregating diverse reasoning paths enhances the stability of predictions and improves outcomes across different levels of model capacity.

6.2 BERTScore Interpretation

Although the models achieve high accuracy in solving mathematical word problems, the corresponding BERTScore values (around 71–72%) remain relatively moderate. This is because BERTScore measures similarity between generated and reference reasoning texts, whereas multiple valid reasoning paths can lead to the same correct answer.

As a result, correct solutions expressed with different wording or structure may receive lower BERTScore values. Therefore, accuracy is a more reliable metric for evaluating task success, while BERTScore serves as a complementary measure of textual similarity. Future work may explore alternative evaluation methods that better capture reasoning validity, such as step-level verification or human evaluation.

6.3 Cross-Lingual Performance Analysis

The results reveal a clear performance gap in solving Arabic mathematical word problems,

particularly for open-source language models. This gap can be partially attributed to the intrinsic limitations of these models, as well as their insufficient exposure to Arabic during training, where logical reasoning patterns are predominantly learned from English data. Open-source models also suffer from a scarcity of Arabic mathematical datasets, weaker representations of coherent step-by-step reasoning, and limited alignment during training. Furthermore, the morphological and syntactic complexity of the Arabic language introduces additional challenges for effective encoding and semantic understanding of mathematical text. As a result, models often rely on implicit internal translation before solving Arabic problems, which increases the likelihood of errors and leads to weaker performance compared to English. Open-source models generally underperform due to insufficient exposure to Arabic mathematical reasoning during training, highlighting the importance of fine-tuning Arabic-focused models.

In contrast, closed-source OpenAI models achieve stronger performance than open-source counterparts, largely due to their training on broader and higher-quality datasets that include mathematical content in both Arabic and English. This multilingual exposure enhances their ability to comprehend mathematical expressions and generate more accurate solutions across languages.

Nevertheless, English remains the primary language in which logical reasoning and problem-solving capabilities are learned, reflecting the dominance of high-quality mathematical resources in English relative to Arabic.

Performance differences among OpenAI models further reflect variations in model architecture and training objectives: GPT-4o demonstrates strong overall performance due to its exposure to high-quality data, while O3-mini is specifically optimized for logical reasoning tasks.

These factors explain the consistently superior performance of these models in solving mathematical problems in both Arabic and English.

6.4 Dataset Characteristics

The three datasets revealed different aspects of Arabic mathematical reasoning. These datasets differ in the difficulty levels, and this resulted in varying performance of the LLMs. AGSM8K and Qudurat datasets are considered medium level of difficulty, while ArabicMWPs is generally regarded as a simpler dataset. AGSM8K provided comprehensive coverage of grade-school mathematics, Qudurat reflected real academic assessment challenges, and ArabicMWPs offered basic arithmetic baselines. This diversity enables thorough evaluation of model capabilities.

Figure 3 illustrates accuracy across all three datasets with GPT-3.5, GPT-4o-mini, ALLaM, and O3-mini when using zero-shot CoT. ArabicMWPs achieved the highest accuracy.

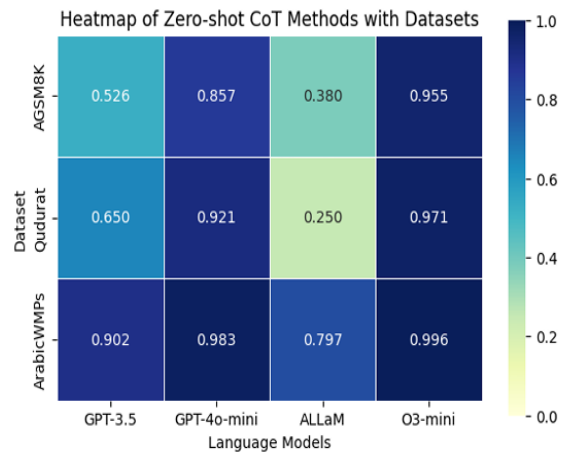


Figure 3: Accuracy of Datasets

7. Limitation and Future Work

Discussion of the limitations is important because it provides a broad understanding of the difficulties encountered in this research and helps in thinking about finding solutions to these problems in the future. There are several limitations that must be considered:

(1) Computational costs associated with extensive experimentation. A significant limitation of this research concerns the costs, especially associated with OpenAI LLMs. Using these models requires payment of costs which is calculated based on the number of tokens. The cost of each LLM is different from the other, in addition to the cost of training the models. In addition, the price of each prompting method differs from one to another due to the length of the prompt and API calls.

(2) Time when creating the new datasets. One of the difficulties in this research was the long time required to translate the data because of the large number of questions translated. In addition, the translation process needed to be accurate, as a

translation error could lead to misunderstanding of the question and resulting in an incorrect answer.

(3) Time when finding the answer. There are some cases that take a long time to find the result, such as self-consistency with high number of samples in open-source LLMs. Moreover, fine tuning LLMs also takes time for open-source models compared with closed-source models.

(4) Resource restrictions on Google Colab necessitated selecting smaller open-source models. Colab allows users to download the LLM in notebook with the proper code to use it. However, resources provided by Colab are not enough for some open-source LLM with big sizes of parameters, for example: 30B or 70B.

(5) Self-consistency results were not reported for the ALLaM model due to the high computational cost and latency associated with generating multiple reasoning paths. Running self-consistency with large sample sizes was not practical within the available resources. Therefore, results based on CoT and Zero-shot CoT prompting techniques are reported, as they still provide meaningful insights into the model's performance.

Future work should explore: (1) Development of larger Arabic mathematical datasets, (2) Investigation of Arabic-specific prompting strategies, (3) fine-tuning more LLMs on the AGSM8K dataset, and (4) Integration with Arabic mathematical education systems.

8. Conclusion

LLMs have successfully performed most NLP tasks, both simple and complex. Solving MWPs requires reasoning thinking to determine the correct answer. Prompt engineering constitutes a fundamental concept that has emerged in recent times with the aim of improving the understanding of LLMs for difficult tasks without the need to train the model. This work represents the first comprehensive evaluation of prompting techniques for Arabic mathematical word problem solving. The introduction of three Arabic datasets and the systematic evaluation of LLMs using three prompting techniques provide valuable insights for the Arabic NLP community.

Key findings include: (1) Self-consistency with GPT-4o achieves state-of-the-art performance (97.65%) on Arabic MWPs, (2) Fine-tuning significantly improves open-source model performance, (3) Substantial performance gaps exist between English and Arabic mathematical reasoning especially in open-source models, and (4) Arabic-specific datasets and evaluation frameworks are essential for progress in Arabic NLP. These results establish important benchmarks for Arabic mathematical reasoning

and provide resources for future research in this critical domain.

9. Acknowledgments

This research was funded by the College of Graduate Studies at Kuwait University. The authors gratefully acknowledge their support, which made this research possible.

10. Bibliographical References

Ahn, J., Verma, R., Lou, R., Liu, D., Zhang, R., and Yin, W. (2024). Large language models for mathematical reasoning: Progresses and challenges. arXiv:2402.00157.

Albahli, S. (2025). An advanced natural language processing framework for Arabic named entity recognition: A novel approach to handling morphological richness and nested entities. *Applied Sciences*, 15(6), 3073.

Alghamdi, R., Liang, Z., and Zhang, X. (2022). ArMath: A dataset for solving Arabic math word problems. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)*, pages 351–362.

Al-Khalifa, S., and Al-Khalifa, H. (2024). The Qiyas benchmark: Measuring ChatGPT mathematical and language understanding in Arabic. arXiv:2407.00146.

Allam, A., Ahmed, S., Hamdi, A., and Mohammed, A. (2025). Arabic large language models for medical text generation. In *Proceedings of the 4th International Conference on Computer Technologies (ICCTech)* (pp. 1–6).

Amatriain, X. (2024). Prompt design and engineering: Introduction and advanced methods. arXiv:2401.14423.

Bari, M. S., Alnumay, Y., Alzahrani, N. A., Alotaibi, N. M., Alyahya, H. A., AlRashed, S. et al. (2024). Allam: Large language models for Arabic and English. arXiv:2407.15390.

Basem, M., Oshallah, I., Hamdi, A., and Mohamed, A. (2025). Few-shot prompting for extractive Quranic QA with instruction-tuned LLMs. In *Proceedings of the Intelligent Methods, Systems, and Applications (IMSA)* (pp. 24–29).

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P. et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L. et al. (2021). Training verifiers to solve math word problems. arXiv:2110.14168.

Diao, S., Wang, P., Lin, Y., Pan, R., Liu, X., and Zhang, T. (2023). Active prompting with chain-of-

thought for large language models. arXiv:2302.12246.

Hosseini, M. J., Hajishirzi, H., Etzioni, O., and Kushman, N. (2014). Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 523–533).

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. arXiv:2205.11916.

Kuwait University. Math Test Samples. Available at: <https://cem.ku.edu.kw/mathsample>

Kwon, S. Y., Bhatia, G., Nagoud, E. M. B., and Abdul-Mageed, M. (2023). ChatGPT for Arabic grammatical error correction. arXiv:2308.04492.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9), 1–35.

Lu, P., Qiu, L., Yu, W., Welleck, S., and Chang, K.-W. (2022). A survey of deep learning for mathematical reasoning. arXiv:2212.10535.

Roy, S., and Roth, D. (2016). Solving general arithmetic word problems. arXiv:1608.01413.

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K. W., and Lim, E. P. (2023). Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. arXiv:2305.04091.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S. et al. (2022). Self-consistency improves chain of thought reasoning in language models. arXiv:2203.11171.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F. et al. Chain-of-thought prompting elicits reasoning in large language models. arXiv:2201.11903.

World Population Review (2024). Arabic speaking countries. Available at: <https://worldpopulationreview.com/country-rankings/arabic-speaking-countries>

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 11809–11822.

Zhang, Q., Chen, M., Bukharin, A., Karampatziakis, N., He, P., Cheng, Y. et al. (2023). AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. arXiv:2303.10512.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). BERTScore: Evaluating text generation with BERT. arXiv:1904.09675.

DIA2: A Comprehensive and Diverse Diacritized Modern Standard Arabic Corpus for Large-Scale NLP Research

Fatima Dekmak¹, Shady Elbassuoni¹, Khaled Bashir Shaban²
Hazem Hajj¹, Wassim El Hajj³, Yasmine Abu Adla¹, Buthaina Alabrash¹

¹ American University of Beirut, Lebanon

² Qatar University, Doha, Qatar

³ American University of Beirut – Mediterraneo, Pafos, Cyprus

fkd04@mail.aub.edu, se58@mail.aub.edu, khaled.shaban@qu.edu.qa,
hh63@mail.aub.edu, wassim@aubmed.ac.cy, yaa41@mail.aub.edu,
bga07@mail.aub.edu

Abstract

The development of Arabic natural language processing (NLP) applications and large language models (LLMs) faces substantial challenges, primarily due to the scarcity of high-quality native Arabic datasets. To address this critical gap, we present DIA2 (a Comprehensive and Diverse Diacritized Modern Standard Arabic Corpus), a novel dataset curated from 28 diverse, carefully selected Arabic sources. DIA2 emphasizes the use of original Arabic text and explicitly avoids machine-translated content. The corpus incorporates substantial amounts of text from books, news articles, and poetry, and employs extensive data preprocessing to support NLP research and LLM development. Our preprocessing pipeline includes rigorous text cleaning, URL- and document-level deduplication, and automatic diacritization, while preserving a gold diacritized subset derived from manually annotated sources. The resulting corpus comprises over 140 GB of high-quality text, containing more than 26 million unique words and 41.9 billion tokens. To evaluate the proposed pipeline, we conducted controlled continued pretraining experiments using Llama3.1-8B on both raw and processed subsets of DIA2. The model trained on processed data consistently outperformed its counterpart across multiple Arabic evaluation benchmarks. These results highlight the positive impact of systematic preprocessing and the utility of DIA2 in empowering native Arabic LLMs and downstream NLP tasks.

Keywords: Dataset, Modern Standard Arabic, Large Language Models, Natural Language Processing, Preprocessing

1. Introduction

Arabic is one of the most widely spoken and culturally significant languages in the world, spoken by over 422 million people across 22 countries in the Middle East and North Africa (Habash, 2010; Farghaly and Shaalan, 2009). Despite rapid progress in natural language processing (NLP) for many high-resource languages, Arabic research remains comparatively underdeveloped, largely due to a combination of *morphological richness* (Habash, 2010; Farghaly and Shaalan, 2009), *dialectal diversity*, *orthographic variation* (El-Haj et al., 2014; Ibrahim et al., 2019; Ali, 2018), and limited availability of *large-scale, high-quality linguistic resources* (Darwish, 2018; Zaghouani, 2014; Alshaikh et al., 2021). As a result, most recent advances in NLP and large language models (LLMs) continue to disproportionately benefit high-resource languages (Wang et al., 2018; Ruder, 2018), leaving Arabic under-represented and constrained by fragmented, inconsistently curated, and often inaccessible datasets (Darwish, 2018; Zaghouani, 2014; Alshaikh et al., 2021).

Recent efforts have begun to address this gap through the release of large-scale Arabic corpora. However, many existing resources rely heavily on

web-crawled or Common Crawl data, which can introduce substantial noise, duplication, and domain imbalance (Aloui et al., 2024; Farhat et al., 2024; Kreutzer et al., 2022). Other datasets incorporate machine-translated text (Sengupta et al., 2023; Ali et al., 2023), potentially distorting Arabic linguistic structure through cross-lingual artifacts. Moreover, most large Arabic corpora omit diacritics (Zerrouki and Balla, 2017; Fadel et al., 2019; Farghaly and Shaalan, 2009), despite their critical role in preserving syntactic, morphological, and semantic distinctions in Arabic, a morphologically rich language where meaning is often ambiguous without vocalization. These limitations collectively restrict the effectiveness of Arabic datasets for training robust language models and downstream NLP systems.

To address these challenges, we introduce DIA2 (a Comprehensive and Diverse Diacritized Modern Standard Arabic Corpus), an open-source, large-scale Arabic dataset explicitly designed to support high-quality NLP research and LLM development. Unlike many existing corpora, DIA2 is curated exclusively from native Arabic sources, avoiding reliance on machine-translated data. The dataset draws from 28 carefully selected sources, including books, news articles, encyclopedic content, and poetry, to ensure domain diversity and linguistic authentic-

ity. DIA2 is further distinguished by rigorous and transparent preprocessing pipeline, incorporating extensive data cleaning, URL- and document-level deduplication, and automatic diacritization, while preserving a gold diacritized subset derived from manually annotated resources¹.

The resulting corpus comprises over 140 GB of high-quality diacritized Arabic text, containing more than 26 million unique words and 41.9 billion tokens, making DIA2 one of the largest publicly available diacritized resources for Modern Standard Arabic (MSA)². In addition to the diacritized version, a parallel non-diacritized variant is also released³, enabling flexible use across a wide range of NLP tasks. To assess the practical utility of DIA2, we conduct controlled continued pretraining experiments using LLaMA 3.1 (8B) (Grattafiori et al., 2024).

2. Related Datasets

Large-scale Arabic corpora have recently emerged to address the scarcity of extensive datasets for Arabic language modeling. For instance, the 101 Billion Arabic Words Dataset (Aloui et al., 2024) targets authentic Arabic content and applies a rigorous cleaning pipeline to reduce noise and duplication. Similarly, the 1.5 Billion Words Arabic Corpus (El-khair, 2016), which is primarily composed of news articles, has been widely adopted in Arabic NLP research, although its scale remains limited for training contemporary LLMs.

More recent efforts aim to expand both scale and domain coverage. The Large Arabic Corpus for LLMs (Ali et al., 2023) integrates books and news sources, but also includes lower-quality web-scraped and machine-translated content, raising concerns about noise, mistranslations, and linguistic inconsistencies. Similarly, the Jais Dataset (Sengupta et al., 2023) aggregates large volumes of Arabic text but relies partly on translated and web-crawled data, potentially introducing cross-lingual artifacts and domain imbalance.

Other initiatives focus on refining Arabic-centric resources. AraMUS (Alghamdi et al., 2023) expands monolingual Arabic data by aggregating and preprocessing existing corpora, emphasizing scale and model training efficiency. While effective for large-scale pretraining, such approaches inherit limitations from their source datasets, including inconsistent cleaning and limited linguistic control.

¹<https://huggingface.co/datasets/DIA2-Arabic/DIA2-Tashkeela-Gold>

²<https://huggingface.co/datasets/DIA2-Arabic/DIA2-Dataset-Diacritized>

³<https://huggingface.co/datasets/DIA2-Arabic/DIA2-Dataset-Non-Diacritized>

In parallel, several task-specific Arabic datasets have been released, including KIND (Yamani et al., 2024), ArQuAD (Obeidat et al., 2024), CIDAR (Alyafeai et al., 2024), and ArabicaQA (Abdallah et al., 2024). These resources provide high-quality annotations for targeted tasks such as classification, reading comprehension, and question answering, but are narrow in scope and not designed for large-scale language model pretraining.

In summary, existing Arabic datasets contribute valuable resources across different scales and tasks; however, many suffer from one or more limitations, including heavy reliance on web-crawled data, inclusion of machine-translated text, lack of systematic deduplication, omission of diacritics, or restricted domain coverage. These gaps underscore the need for large-scale, high-quality, natively sourced, and consistently curated Arabic corpora. DIA2 directly addresses these limitations by combining native Arabic sources, extensive preprocessing, large-scale diacritization, and public availability, positioning it as a complementary and foundational resource for advancing Arabic NLP research and language modeling.

3. Dataset Overview

DIA2 is a large-scale corpus focused on MSA, designed to address the persistent lack of high-quality, systematically curated Arabic datasets required for reliable NLP research and language model training (Aloui et al., 2024). The dataset was curated from 28 publicly available native Arabic sources, ensuring broad domain coverage and linguistic authenticity, while explicitly avoiding any machine-translated content (Obeidat et al., 2024). The sources span a broad geographic range, covering the Gulf (Oman, Saudi Arabia), the Levant, Egypt, North Africa (Morocco), as well as multi-regional resources, alongside classical Arabic resources representing the pre-Islamic and early Islamic periods. As shown in Table 1, the data sources are distributed across general-purpose content (72.93%), news (26.23%), books (0.50%), and poetry (0.35%).

The general-purpose category comprises diverse text types suitable for large-scale language modeling, including *CulturaX* (Nguyen et al., 2023), *NADiA* (Al-Debsi et al., 2019), *SANAD* (Einea et al., 2019), and *Arwiki* (Contributors, 2024).

The news category focuses on contemporary topics and includes several Arabic news corpora such as *AraNews* (Nagoudi et al., 2020), *Arabic News Articles* (Altamimi and Alayba, 2023), the *Ultimate Arabic News Dataset* (Al-Dulaimi, 2022), and the *Saudi Newspapers Corpus* (Alhagri, 2015).

DIA2 also incorporates formal and literary Arabic, including text from the *Hindawi Bookset* (Elfili, 2023) and the *King Saud University Corpus of*

General Purpose		News	
Source	Size (MB)	Source	Size (MB)
CulturaX (Nguyen et al., 2023)	190,000	NADIA (Al-Debsi et al., 2019)	21,100
Opus Wiki (Wenzek et al., 2020)	29,355	AraNPCC (Al-Thubaity et al., 2022)	20,000
ArWiki (Contributors, 2024)	10,600	SANAD (Einea et al., 2019)	19,100
Tashkeela (Zerrouki and Balla, 2017)	1,220	Arabic News Articles (Altamimi and Alayba, 2023)	17,500
KALIMAT (El-Haj and Koulali, 2013)	511	OSIAN (Zeroual et al., 2019)	6,490
KSUCCA (Alrabiah, 2014)	441	Arabic-News (Saad, 2019)	4,130
Al-Watan (Abbas et al., 2011)	28	Arabic Billion Words (Saad and Ashour, 2019)	1,800
Opus100 (Zhang et al., 2020)	3.49	Arabic Fake News (Khalil et al., 2022)	1,640
Arabic Quotes (Boulahia, 2023)	1.12	Goud-sum (Issam and Mrini, 2022)	622
		AraNews (Nagoudi et al., 2020)	632
		Ultimate Arabic News (Al-Dulaimi, 2022)	543
		Arabic Classification (Mohamed, 2018)	374
		Saudi Newspapers Corpus (Alhagri, 2015)	103
Books		Poems	
Source	Size (MB)	Source	Size (MB)
Sanadset (Mghari et al., 2022)	1,330	Arabic Poems 2 (Yousef et al., 2019)	528
Hindawi Books (Elfilali, 2023)	472	Arabic Poems 1 (Rezk, 2023)	438
		Ashaar (ARBML Team, 2024)	252
		Classical Arabic Poetry (El Karef, 2022)	49

Table 1: Raw Dataset Composition by Category and Source

Classical Arabic (KSUCCA) (Alrabiah, 2014). In addition, the dataset covers poetry and expressive language, drawing from the *Arabic Poem Comprehensive Dataset (Parts 1 & 2)* (Rezk, 2023; Yousef et al., 2019), *Ashaar* (ARBML Team, 2024), and *Classical Arabic Poetry* (El Karef, 2022).

A strict and multilingual-aware preprocessing pipeline was applied to remove noisy, duplicated, and sensitive content while preserving Arabic linguistic structure. The resulting corpus consists primarily of MSA, with limited inclusion of Classical Arabic to support stylistic and historical linguistic diversity. To address the scarcity of diacritized Arabic resources, DIA2 provides automatically diacritized text generated using the CATT model (Alasmary et al., 2024). In parallel, DIA2 preserves a gold diacritized subset derived from the manually annotated Tashkeela corpus (Fadel et al., 2019). Deduplication is applied at both the URL and document levels to ensure dataset uniqueness.

DIA2 is designed to support a wide range of Arabic NLP applications, including language modeling, diacritics restoration, text-to-speech, named entity recognition, and MSA versus dialect classification. Its scale, domain diversity, consistent preprocessing, and high-quality diacritization make DIA2 a robust monolingual resource for training and evaluating Arabic language technologies.

4. Dataset Preprocessing

To construct a high-quality Arabic corpus suitable for training modern NLP systems, comprehensive preprocessing is essential to remove noisy, redun-

dant, and low-quality content. Such noise may arise from HTML tags, embedded JavaScript code, boilerplate text, or linguistic anomalies such as excessive punctuation, non-Arabic scripts, and unintelligible character sequences. These artifacts can disrupt Arabic linguistic structure and negatively impact model learning, making their removal a critical step. Accordingly, the proposed preprocessing pipeline in Figure 1 applies a sequence of filtering, cleaning, and normalization stages, including URL-based filtering, deduplication, and metric-based quality control. To support reproducibility and facilitate future research, the full preprocessing pipeline is made publicly available⁴.

4.1. Data Cleaning

Inspired by prior large-scale Arabic preprocessing efforts (Ghaddar et al., 2022; Sengupta et al., 2023), we implemented a fine-grained, sentence-level data cleaning pipeline aimed at systematically eliminating noise while preserving linguistically meaningful content.

4.1.1. HTML and JavaScript Removal

HTML markup and JavaScript code are common artifacts in open-source datasets derived from web sources and can significantly interfere with downstream text processing. These elements were removed using regular expressions (Regex) and the

⁴<https://github.com/fatimadekmak/DIA2-arabic-preprocessing>

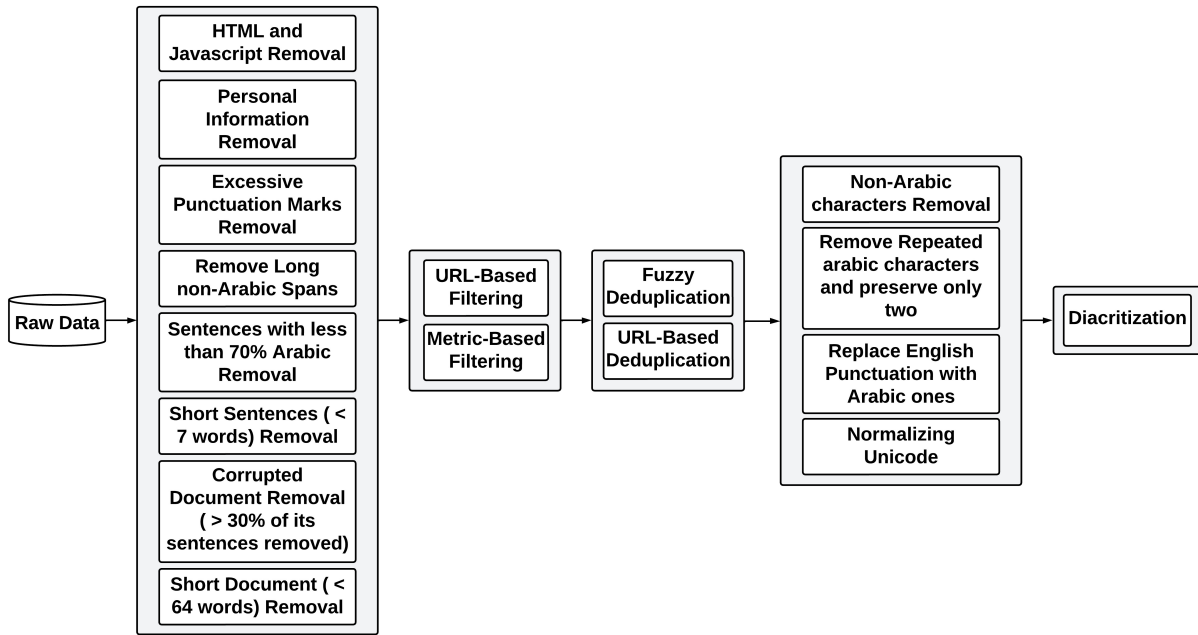


Figure 1: Data Preprocessing Pipeline

BeautifulSoup library⁵. RegEx patterns were utilized to identify and eliminate JavaScript segments (e.g., `<script>...</script>`) while BeautifulSoup was used to extract clean textual content from HTML documents, ensuring robust separation between text and markup.

4.1.2. Personal Information Removal

Open-source corpora may inadvertently contain personal information, raising privacy and ethical concerns. DIA2 mitigates this risk through targeted anonymization, replacing identifiable patterns such as phone numbers and email addresses with standardized placeholders (e.g., +999-999-9999 and Example@mail.com). Phone numbers are detected using RegEx patterns, while the phonenumbers Python library⁶ was employed to improve detection robustness across formats.

4.1.3. Excessive Punctuation Marks Removal

Excessive punctuation can introduce artificial patterns and reduce textual coherence. DIA2 applies regular-expression-based filtering to remove punctuation sequences exceeding three consecutive characters, thereby improving readability while preserving legitimate expressive usage.

⁵<https://pypi.org/project/beautifulsoup4/>

⁶<https://pypi.org/project/phonenumbers/>

4.1.4. Enhancing Language Consistency

To maintain Arabic linguistic integrity, two complementary filtering strategies were applied. Regular expressions were used to identify and remove long non-Arabic spans, while selectively preserving essential elements such as numerals and commonly used symbols. This step also helped eliminate residual artifacts such as URLs, markup fragments, and code snippets that persisted after HTML and JavaScript removal. In addition, sentences containing less than 70% Arabic characters were removed to ensure that retained content is predominantly Arabic, effectively filtering out multilingual noise and irrelevant text.

4.1.5. Ensuring Data Quality

Additional quality constraints were applied to improve dataset coherence. Sentences shorter than eight words were removed due to limited informational value, and documents containing fewer than 64 words were discarded to ensure sufficient contextual content. Furthermore, documents in which more than 30% of sentences were removed during preprocessing were considered fragmented and excluded.

4.2. URL-Based Filtering

To ensure content appropriateness and ethical alignment, URL-based filtering was employed to exclude sources associated with unsafe content. We

utilized the UT15 blacklist⁷, which has been widely adopted in large-scale dataset filtering (Nguyen et al., 2023; Penedo et al., 2023).

4.3. Metric-Based Filtering

Metric-based filtering was applied to identify and remove gibberish and low-quality text using a set of heuristic criteria. These included excessive character repetition, minimum Arabic word thresholds, validation against common Arabic vocabulary, and constraints on Arabic character proportions. Text containing non-Arabic scripts, such as Persian (Farsi), was further filtered using language identification tools (e.g., langdetect⁸). Collectively, these steps removed up to 35% of noisy entries, substantially improving dataset quality, consistent with prior large-scale preprocessing findings.

4.4. Arabic-Specific Cleaning Steps

Beyond general filtering, Arabic-specific normalization steps were applied to ensure linguistic consistency and suitability for Arabic NLP tasks.

4.4.1. Removing Non-Arabic Characters

An allowlist of valid characters was defined, including Arabic script, Arabic and Latin numerals, punctuation, and diacritics. All other characters were removed to maintain Arabic-centric textual integrity.

4.4.2. Removing Repeated Arabic Characters

Character repetition is common in Arabic, often used for emphasis or stylistic expression. While previous work removed all repeated characters (Hegazi et al., 2021; Mubarak and Darwish, 2014), our manual inspection revealed that many valid Arabic words naturally contain double character occurrences. To balance normalization and linguistic authenticity, DIA2 allows up to two consecutive occurrences of the same character, preventing excessive repetition while avoiding semantic distortion.

4.4.3. Punctuation and Unicode Normalization

Following (Sengupta et al., 2023), English punctuation marks were replaced with their Arabic counterparts to ensure textual consistency. For instance, the question mark (?) was replaced with (؟), the semicolon (;) with (؛), and the comma (,) with (،). In addition, Unicode normalization was applied to

ensure consistent representation of Arabic characters, reducing encoding variability and simplifying downstream processing (Aloui et al., 2024).

4.5. Diacritization

While most Arabic corpora remove diacritics to simplify representation, this practice discards linguistically critical information. Given the objectives of DIA2, diacritics were explicitly restored to preserve syntactic and semantic distinctions.

A comparative evaluation was conducted to select an appropriate diacritization system, comparing CAMEL⁹ and CATT (Character-based Arabic Tashkeel Transformer) (Alasmary et al., 2024). We adopted the encoder-only variant of CATT, which offers substantially faster inference with minimal performance degradation, making it suitable for large-scale processing.

Since CATT is a character-level model trained on clean MSA text, out-of-distribution inputs such as raw digits and non-MSA characters are incompatible. Accordingly, Persian-origin characters are normalized to their MSA equivalents and numeric tokens are expanded to Arabic word forms via num2words¹⁰. Original digits are restored post-diacritization through index mapping. Sequences exceeding 1,024 characters are segmented at punctuation boundaries.

4.5.1. Evaluation Setup

We sampled 150 sentences from DIA2 and restored diacritics using both tools, resulting in 300 sentences for evaluation. Three native Arabic annotators participated, each reviewing 200 sentences, with 100 sentences shared across annotators to enable inter-annotator agreement analysis.

Annotators were presented with a random mix of outputs and manually counted the number of incorrect diacritics, incorrect case-ending diacritics (الإعراب), and incorrectly diacritized words. Annotators also reported confidence for each judgment, which was later aggregated as an indicator of annotation reliability.

4.5.2. Inter-Annotator Agreement

Error counts were discretized into five bins: 0, 1, 2–4, 5–7, and 8+. This binning reflects error severity rather than precise counts and is suitable for comparative evaluation. Inter-annotator agreement was measured using Krippendorff’s α (ordinal) (Krippendorff, 2011), which supports multiple annotators and does not assume equal distances

⁷<https://dsi.ut-capitole.fr/blacklists/>

⁸<https://pypi.org/project/langdetect/>

⁹<https://camel-tools.readthedocs.io/en/latest/api/disambig/mle.html>

¹⁰<https://pypi.org/project/num2words/>

between categories. As shown in Table 2, agreement scores ranged from 0.676 to 0.705 across the three evaluated dimensions, indicating acceptable agreement for exploratory and comparative analysis.

Annotator self-reported confidence rates were high overall (all above 92%), indicating that the task was well defined and the annotations are reliable for comparative evaluation.

Metric	Krippendorff's α	Interpretation
Incorrect Diacritics	0.705	Good
Incorrect Case Endings	0.680	Good
Incorrectly Diacritized Words	0.676	Good

Table 2: Inter-Annotator Agreement Scores (Krippendorff's α (Ordinal))

4.5.3. Tool Comparison Results

Using the human annotations, we computed four evaluation metrics: diacritic error rate, case-ending error rate, word error rate, and diacritization coverage. The results are shown in Table 3. Across all error-based metrics, CATT consistently produced fewer errors than CAMEL, while also achieving near-complete coverage of diacritized words. In contrast, CAMEL left a larger proportion of words undiacritized. Based on these results, CATT was selected as the diacritization tool for DIA2. All automatically diacritized versions of DIA2 released therefore rely on CATT Encoder-only outputs.

Metric	CAMEL	CATT
Diacritic error rate ↓	0.095	0.036
Case-ending error rate ↓	0.231	0.081
Word error rate ↓	0.269	0.125
Coverage rate ↑	0.936	0.998

Table 3: Diacritization Performance Comparison Between CAMEL and CATT

4.6. Deduplication

To prevent redundancy and reduce memorization effects during training, deduplication was applied at both URL and document levels. Duplicate entries originating from the same URL were removed by retaining a single instance per source. Since URLs were available for only a subset of the data, document-level deduplication was additionally applied. This step was performed using the datasketch¹¹ library, utilizing MinHash and Locality-Sensitive Hashing (LSH) for efficient near-duplicate

¹¹<https://pypi.org/project/datasketch/>

detection. A sharding strategy inspired by (Öhman et al., 2023) enabled scalable intra- and inter-shard deduplication. The process was controlled by two parameters: the number of permutations for similarity estimation and a Jaccard similarity threshold for duplicate detection. The rationale behind these choices is discussed in Appendix B. Overall, this process reduced the dataset size by approximately 25%, from 70,522,230 to 52,449,241 rows.

5. Dataset Evaluation

5.1. Dataset Size Reduction

The cumulative effect of the processing steps described in the previous section resulted in a substantially more compact and higher-quality, with the final version being 63.5% smaller than the original raw data. This reduction reflects the systematic removal of noisy, duplicated, and low-quality content, rather than indiscriminate filtering.

As illustrated in Figure 2, the dataset size decreased progressively across preprocessing stages. Starting from an initial size of approximately 329.26 GB, the data volume was reduced after each step, reaching a minimum of around 94.68 GB following deduplication. The subsequent diacritization stage increased the dataset size, due to the addition of vocalization marks, resulting in a final corpus size of approximately 140 GB.

In terms of instance counts, the dataset was reduced from 115,895,312 rows to 52,449,241 rows. Of the retained data, approximately 80% consists of long-form text (e.g., articles and book chapters), while the remaining 20% corresponds to shorter texts, including poems, quotations, and brief passages. The final corpus contains 18,376,574,554 words and 26,266,877 unique words. The average document length is 17 sentences, with an average sentence length of 26 words. Tokenization using the LLaMA 3.1-8B tokenizer (Grattafiori et al., 2024) yields a total of **41,931,425,578** tokens.

In addition to the automatically diacritized corpus, DIA2 includes a clean gold diacritized subset derived from the Tashkeela corpus (Fadel et al., 2019). This subset comprises **50,000 rows** (around **4.8 million tokens**) preserved with their original manual diacritics. Cleaning this subset resulted in an approximately 9% reduction in size, primarily due to the removal of short or noisy entries.

5.2. Comparison with Other Datasets

Table 4 presents a comparative overview of DIA2 against several widely used Arabic datasets, including 101 Billion Arabic Words (Aloui et al., 2024), the 1.5 Billion Words Arabic Corpus (El-khair, 2016), the Jais Dataset (Sengupta et al., 2023), AraMUS (Alghamdi et al., 2023), and ArabicWeb24 (Farhat

Feature	101BW	1.5BW	LDAC	Jais	Aramus	AW24	DIA2
Dialectal Content	✓	✓	✓		✓		
Diacritization							✓
Free of Translated Data	✓	✓			✓	✓	✓
Data Cleaning	✓		✓	✓	✓	✓	✓
Deduplication	✓			✓	✓	✓	✓
Code Availability						✓	✓
Public Availability	✓	✓				✓	✓
Includes Curated High-Quality Data			✓	✓	✓		✓

Table 4: Quantitative Comparison of DIA2 with other Arabic datasets: 101BW (101 Arabic Billion Words), 1.5BW (1.5 Billion Words Arabic Corpus), LDAC (Large and Diverse Arabic Corpus for LLMs), JAIS (JAIS Dataset), ARAMUS (Aramus Corpus), and AW24 (ArabicWeb24)

Feature	101BW	1.5BW	LDAC	Jais	Aramus	AW24	DIA2
Word Count	101B	1.5B	N/A	72B tokens*	N/A	26B tokens*	18.4B
Unique Words	N/A	3.3M	N/A	N/A	N/A	N/A	26.3M
Total Rows/Texts	89.1M	5.2M	N/A	N/A	N/A	86.8M	52.5M
Data Size	400GB	16GB	500GB	N/A	529GB	199 GB	140GB

Table 5: Quantitative Comparison of DIA2 with other Arabic Datasets

*: Token count was provided rather than the word count.

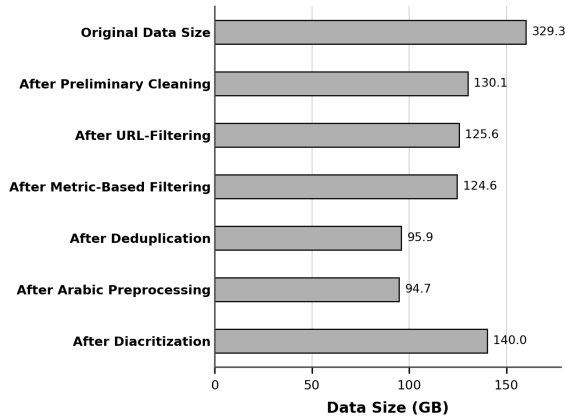


Figure 2: Change in Data Size of DIA2 across Processing Steps

et al., 2024). The comparison highlights key characteristics such as dialectal coverage, diacritization, reliance on translated data, data cleaning and deduplication, as well as code and data availability.

As shown in Table 4, most existing corpora lack diacritization, rely largely on web-crawled or machine-translated content, and offer limited preprocessing transparency or public availability. DIA2 addresses these gaps: it includes curated high-quality data, provides large-scale diacritization as

a distinctive feature, applies rigorous data cleaning and multi-level deduplication, and releases both the dataset and preprocessing pipeline publicly. These characteristics collectively position DIA2 as a complementary, higher-fidelity resource for Arabic language modeling and evaluation.

6. Preprocessing Pipeline Evaluation

To assess the effectiveness of the proposed preprocessing pipeline, we conducted a controlled continued pretraining experiment in which the same base model was trained on raw and preprocessed subsets of DIA2 under identical conditions. The objective of this experiment is not to achieve state-of-the-art performance, but rather to isolate the impact of preprocessing on model behavior. Continued pretraining was performed on a limited subset of the dataset due to computational constraints, while maintaining sufficient scale for reliable relative comparison.

6.1. Pretraining Methodology

We performed continued pretraining of LLaMA 3.1-8B using Low-Rank Adaptation (LoRA), a parameter-efficient fine-tuning approach (Hu et al., 2021). Training was implemented using

Model	ArabicMMLU					Copa Arabic	AraDiCE PIQA MSA	Arabic Leaderboard		TruthfulQA Arabic	
	General	Grammar	High Arabic	Middle Arabic	Primary Arabic			Light	Complete	MC1	MC2
LLaMA-3.1-8B	0.577	0.515	0.344	0.519	0.433	0.629	0.592	0.453	0.458	0.287	0.454
<i>Pretrained on DIA2</i>											
DIA2 (raw)	0.525	0.479	0.328	0.333	0.345	0.618	0.588	0.452	0.454	0.274	0.444
DIA2 (prep.)	0.528	0.482	0.349	0.630	0.389	0.596	0.584	0.457	0.454	0.310	0.476

Table 6: Evaluation results of the original LLaMA-3.1-8B model and its continued pretraining variants on DIA2 (raw and preprocessed subsets) across multiple Arabic NLP benchmarks

Hugging Face Transformers, bitsandbytes for mixed-precision optimization, and accelerate for distributed execution, on a multi-GPU setup consisting of three NVIDIA GeForce RTX 4060 Ti GPUs (16 GB each).

Since the model is already multilingual and pre-trained on diverse corpora, no tokenizer retraining or vocabulary expansion was performed. Continued pretraining was conducted exclusively on a subset of the non-diacritized version of DIA2, ensuring that performance differences stem from data quality rather than representation changes.

The training dataset comprised of 166K rows, with a batch size of 4 and gradient accumulation over 8 steps. The maximum sequence length was set to 512 tokens, and training was conducted for three epochs. LoRA was configured with a rank parameter of 8, using the 8-bit AdamW optimizer and a learning rate of 5×10^{-5} .

6.2. Performance Comparison

Table 6 reports the performance of the original multilingual LLaMA 3.1-8B model alongside its continued-pretraining variants trained on raw and preprocessed DIA2 data. Evaluation was conducted using the lm-eval-harness¹² framework across multiple Arabic benchmarks, including ArabicMMLU (Lakim et al., 2024), the Open Arabic LLM Leaderboard benchmarks (El Filali et al., 2024), and TruthfulQA Arabic (Lakim et al., 2024), covering general knowledge, grammatical competence, reasoning, and task-specific understanding.

Across the majority of benchmarks, the model pretrained on the preprocessed DIA2 subset consistently outperformed the variant trained on raw data. The largest relative improvements were observed on **High Arabic (+0.021)**, **Middle Arabic (+0.297)**, and **MC2 (+0.032)**, suggesting that cleaning and data normalization improve model robustness under resource-constrained continued pretraining.

Notably, performance gains were not uniform across all benchmarks. The preprocessed dataset did not yield improvements on **COPA Arabic** and **AraDiCE PIQA MSA**, both of which are translated

from English. These benchmarks may introduce cross-lingual artifacts and alignment inconsistencies, limiting the benefits of Arabic-specific pretraining and reducing evaluation sensitivity to improvements in native Arabic data quality.

While the original multilingual model performed competitively overall, it showed weaker performance on several Arabic-specific benchmarks, particularly High Arabic and Primary Arabic. In contrast, the model trained on the preprocessed DIA2 subset achieved the highest scores on High Arabic (0.349), Middle Arabic (0.630), and the Arabic Leaderboard (light version) (0.457), indicating improved modeling of Arabic linguistic structure following exposure to cleaner, natively curated data.

Due to computational constraints and limited public availability of comparable datasets, we did not conduct continued pretraining experiments on alternative Arabic corpora listed in Table 4. Nevertheless, the consistent performance advantages observed when using preprocessed DIA2 data provide empirical evidence that rigorous preprocessing materially improves model behavior, even under limited training budgets.

7. Conclusion and Future Work

This work introduces DIA2, a large-scale, natively sourced, and diacritized corpus for MSA, designed to address longstanding limitations in Arabic NLP resources. DIA2 combines broad domain coverage, linguistic authenticity, systematic preprocessing, and large-scale diacritization, resulting in a dataset that supports both foundational language modeling and downstream Arabic NLP tasks. By explicitly avoiding machine-translated content and applying multi-level deduplication and Arabic-specific normalization, DIA2 offers a high-fidelity alternative to existing large-scale Arabic corpora.

We further demonstrated the value of DIA2 through a controlled preprocessing pipeline evaluation, showing that continued pretraining on preprocessed DIA2 subsets outperforms training on raw data across multiple Arabic benchmarks. These results empirically validate the importance of rigorous data curation and preprocessing for under-resourced languages such as Arabic. Notably, im-

¹²<https://github.com/EleutherAI/lm-evaluation-harness>

provements were most pronounced on native Arabic benchmarks, underscoring the limitations of translated evaluation sets for accurately measuring Arabic language understanding.

Looking forward, several directions remain open for future work. First, expanding DIA2 to include additional high-quality dialectal Arabic data, while maintaining strict quality and provenance controls, could further enhance its applicability. Second, scaling continued pretraining to larger portions of the corpus and higher-capacity models would enable a more comprehensive assessment of DIA2’s impact on foundation model performance. Third, systematic ablation studies of individual preprocessing components could provide deeper insight into their relative contributions. Finally, human-in-the-loop evaluation and error analysis, particularly for diacritization quality and semantic fidelity, would complement automated benchmarks and support more nuanced model assessment.

By releasing DIA2 and its preprocessing pipeline, we aim to lower barriers to high-quality Arabic NLP research and serve as a foundational resource for future advances in Arabic language modeling, evaluation, and responsible AI development.

8. Limitations

Despite the scale and rigor of DIA2, several limitations should be acknowledged. First, although the dataset draws from 28 diverse native Arabic sources, its content remains predominantly MSA. While this design choice ensures linguistic consistency and broad applicability, it limits direct coverage of regional Arabic dialects, which are increasingly relevant for real-world NLP applications.

Second, the majority of the corpus is automatically diacritized using the CATT model, and therefore may contain residual diacritization errors, particularly in out-of-context sentences, rare proper nouns, and domain-specific terminology. Although a gold diacritized subset is preserved for reference and evaluation, the scale of manually annotated diacritized data remains limited.

Third, the preprocessing pipeline relies on heuristic thresholds (e.g., minimum Arabic character ratios, sentence and document length constraints), which, while informed by prior work and empirical inspection, may inadvertently remove some valid but unconventional Arabic text, such as short-form expressions, quotations, or stylistically creative content.

Fourth, the evaluation of the preprocessing pipeline is based on continued pretraining under constrained computational resources, using a single base model (LLaMA-3.1-8B) and a limited subset of the dataset. While the results provide controlled evidence of preprocessing benefits, they do

not fully capture the potential impact of DIA2 when used for full-scale pretraining or larger model architectures.

Finally, several Arabic benchmarks used for evaluation are translated from English, which may obscure improvements gained from native Arabic data curation and introduce cross-lingual artifacts that limit interpretability of results. This highlights the broader need for high-quality, natively constructed Arabic evaluation benchmarks.

These limitations point to clear and actionable directions for future work, many of which are already outlined in the preceding section, and do not detract from DIA2’s primary contribution as a large-scale, transparently curated, and publicly available Arabic corpus.

9. Bibliographical References

- M. Abbas, K. Smaili, and D. Berkani. 2011. Evaluation of topic identification methods on arabic corpora. *Journal of Digital Information Management*, 9(5):185–192.
- Abdelrahman Abdallah, Mahmoud Kasem, Mahmoud Abdalla, Mohamed Mahmoud, Mohamed Elkasaby, Yasser Elbendary, and Adam Jatowt. 2024. [Arabicaqa: A comprehensive dataset for arabic question answering](#). arXiv preprint arXiv:2403.17848.
- Abdulmohsen Al-Thubaity, Sakhar Alkheryef, and Alia O. Bahanshal. 2022. [AraNPCC: The Arabic newspaper COVID-19 corpus](#). In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur’an QA and Fine-Grained Hate Speech Detection*, pages 32–40, Marseille, France. European Language Resources Association.
- Faris Alasmary, Orjuwan Zaafarani, and Ahmad Ghannam. 2024. [Catt: Character-based arabic tashkeel transformer](#). arXiv preprint arXiv:2407.03236.
- Asaad Alghamdi, Xinyu Duan, Wei Jiang, Zhenhai Wang, Yimeng Wu, Qingrong Xia, Zhefeng Wang, Yi Zheng, Mehdi Rezagholizadeh, and Baoxing Huai. 2023. [Aramus: Pushing the limits of data and model scale for arabic natural language processing](#). arXiv preprint arXiv:2306.06800.
- Abbas Raza Ali, Muhammad Ajmal Siddiqui, Rema Algunaibet, and Hasan Raza Ali. 2023. [A large and diverse arabic corpus for language modeling](#). arXiv preprint arXiv:2201.09227.

- Ahmed Ali. 2018. Speech technology for arabic. In *Handbook of Speech Processing in Inflectional Languages*. Springer.
- Manel Aloui, Hasna Chouikhi, Ghaith Chaabane, Haithem Kchaou, and Chehir Dhaouadi. 2024. [101 billion arabic words dataset](#). arXiv preprint arXiv:2405.01590.
- Al-Salman AbdulMalik Atwell Eric Alrabiah, Maha. 2014. The design and construction of the 50 million words ksucca king saud university corpus of classical arabic. In *Proceedings of the Second Workshop on Arabic Corpus Linguistics (WACL-2)*, Lancaster University, UK.
- Hana Alshaikh, Muath Nagi, et al. 2021. A survey of arabic language technologies in the big data era. *IEEE Access*.
- Mohammed Altamimi and Abdulaziz M. Alayba. 2023. [Anad: Arabic news article dataset](#). *Data in Brief*, 50:109460.
- Zaid Alyafeai, Khalid Almubarak, Ahmed Ashraf, Deema Alnuhait, Saied Alshahrani, Gubran A. Q. Abdulrahman, Gamil Ahmed, Qais Gawah, Zead Saleh, Mustafa Ghaleb, Yousef Ali, and Maged S. Al-Shaibani. 2024. [Cidar: Culturally relevant instruction dataset for arabic](#). arXiv preprint arXiv:2403.17848.
- Kareem Darwish. 2018. The challenges of social media analytics in the arabic language. In *Social Informatics*. Springer.
- Ali El Filali, Hamza Alobeidli, Cl  mentine Fourrier, Basma El Amel Boussaha, Ruxandra Cojocar, Nathan Habib, and Hakim Hacid. 2024. Open arabic llm leaderboard. <https://huggingface.co/blog/leaderboard-arabic>. Accessed: 2025-02-20.
- Mahmoud El-Haj, Udo Kruschwitz, and Chris Fox. 2014. An arabic corpus of academic texts. In *Proceedings of LREC*.
- Ibrahim Abu El-khair. 2016. [1.5 billion words arabic corpus](#). arXiv preprint arXiv:1611.04033.
- Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019. [Arabic text diacritization using deep neural networks](#). arXiv preprint arXiv:1905.01965.
- Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing*, 8(4).
- Abbas Ghaddar, Yimeng Wu, Sunyam Bagga, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Duan Xinyu, Zhefeng Wang, Baoxing Huai, Xin Jiang, Qun Liu, and Philippe Langlais. 2022. [Revisiting pre-trained language models and their evaluation for arabic natural language understanding](#). arXiv preprint arXiv:2205.10687.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzm  n, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Year, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur   lebi, Patrick

Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily

Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang,

- Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The llama 3 herd of models](#). arXiv preprint arXiv:2407.21783.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool.
- Mohamed Osman Hegazi, Yasser Al-Dossari, Abdullah Al-Yahy, Abdulaziz Al-Sumari, and Anwer Hilal. 2021. [Preprocessing arabic text on social media](#). *Heliyon*, 7:e06191.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). arXiv preprint arXiv:2106.09685.
- Ahmed Ibrahim, Muhammad Abdul-Mageed, et al. 2019. On dialectal variation in arabic social media. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*.
- Abderrahmane Issam and Khalil Mrini. 2022. [Goud.ma: A news article dataset for summarization in moroccan darija](#). In *Proceedings of the 3rd Workshop on African Natural Language Processing*.
- Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroko Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsudeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhlov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. [Quality at a glance: An audit of web-crawled multilingual datasets](#). *Transactions of the Association for Computational Linguistics*, 10:50–72.
- Klaus Krippendorff. 2011. [Computing krippendorff's alpha-reliability](#). Annenberg School for Communication, University of Pennsylvania.
- Imad Lakim, Ebtesam Almazrouei, Ibrahim Abualhaol, Merouane Debbah, and Julien Launay. 2024. [Allam: Large language models for arabic and english](#). arXiv preprint arXiv:2407.15390.
- Mohammed Mghari, Omar Bouras, and Abdelaziz El Hibaoui. 2022. [Sanadset 650k: Data on hadith narrators](#). Mendeley Data.
- Hamdy Mubarak and Kareem Darwish. 2014. [Using Twitter to collect a multi-dialectal corpus of Arabic](#). In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 1–7, Doha, Qatar. Association for Computational Linguistics.
- El Moatez Billah Nagoudi, AbdelRahim Elmadany, Muhammad Abdul-Mageed, Tariq Alhindi, and Hasan Cavusoglu. 2020. Machine generation and detection of arabic manipulated and fake news. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 69–84, Barcelona, Spain.
- Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. [Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages](#). arXiv preprint arXiv:2309.09400.
- Rasha Obeidat, Marwa Al-Harbi, Mahmoud Al-Ayyoub, and Luay Alawneh. 2024. [Arquad: An expert-annotated arabic machine reading comprehension dataset](#). *Cognitive Computation*, 16:1–20.
- Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli,

Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. [The refined-web dataset for falcon llm: Outperforming curated corpora with web data, and web data only](#). arXiv preprint arXiv:2306.01116.

Sebastian Ruder. 2018. [Nlp’s imagenet moment has arrived](#). The Gradient.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, and et al. 2023. [Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models](#). arXiv preprint arXiv:2308.16149.

Alex Wang, Amanpreet Singh, et al. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of ICLR*.

Asma Yamani, Raghad Alziyady, Reem AlYami, Salma Albelali, Leina Albelali, Jawharah Almulhim, Amjad Alsulami, Motaz Alfarraj, and Rabeah Al-Zaidy. 2024. [The KIND dataset: A social collaboration approach for nuanced dialect data collection](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, St. Julian’s, Malta. Association for Computational Linguistics.

Waleed A. Yousef, Omar M. Ibrahime, Taha M. Madbouly, and Moustafa A. Mahmoud. 2019. [Learning meters of arabic and english poems with recurrent neural networks: a step forward for language understanding and synthesis](#). arXiv preprint arXiv:1905.05700.

Wajdi Zaghouani. 2014. Critical survey of the freely available arabic corpora. In *Proceedings of LREC*.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. [Osian: Open source international arabic news corpus - preparation and integration into the clarin-infrastructure](#). In *Proceedings of the Fourth Arabic Natural Language Processing Workshop* (pp. 175-182). Association for Computational Linguistics.

Taha Zerrouki and Amar Balla. 2017. [Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems](#). *Data in Brief*, 11:147–151.

Joey Öhman, Severine Verlinden, Ariel Ekgren, Amaru Cuba Gyllensten, Tim Isbister, Evangelia Gogoulou, Fredrik Carlsson, and Magnus Sahlgren. 2023. [The nordic pile: A 1.2tb nordic](#)

[dataset for language modeling](#). arXiv preprint arXiv:2303.17183.

10. Language Resource References

Al-Debsi, Ridhwan and Elnagar, Ashraf and Einea, Omar. 2019. [NADIA: News Articles Dataset in Arabic for Multi-Label Text Categorization](#). Mendeley Data.

Al-Dulaimi, Ahmed Hashim. 2022. [Ultimate Arabic News Dataset](#). Mendeley Data.

Alhagri, M. 2015. [Saudi Newspapers Arabic Corpus \(SaudiNewsNet\)](#). GitHub.

ARBML Team. 2024. [Ashaar](#). HuggingFace.

Boulahia, Ahmed Khalil. 2023. [Arabic Quotes Dataset](#). GitHub. Licensed under Apache License 2.0.

Wikipedia Contributors. 2024. [Arabic Wikipedia Dump](#). Wikimedia Foundation. Retrieved on February 2025.

Einea, Omar and Elnagar, Ashraf and Al-Debsi, Ridhwan. 2019. [SANAD: Single-Label Arabic News Articles Dataset for Automatic Text Categorization](#). Mendeley Data.

El-Haj, Mahmoud and Koulali, Rim. 2013. [KALIMAT: A Multipurpose Arabic Corpus](#). Semantic Scholar.

El Karef, Nehal. 2022. [Classical Arabic Poetry Dataset](#). HuggingFace.

Elfilali, Ali. 2023. [Hindawi Books Dataset](#). HuggingFace.

Farhat, May and Taghadouini, Said and Hallström, Oskar and Hajri-Gabouj, Sonja. 2024. [ArabicWeb24: Creating a High Quality Arabic Web-only Pre-training Dataset](#). LightOn, INSAT.

Khalil, Ashwaq and Jarrah, Moath and Aldwairi, Monther. 2022. [Arabic Fake News Dataset \(AFND\)](#). Mendeley Data.

Mohamed. 2018. [DataSet for Arabic Classification](#). Mendeley Data.

Rezk, Abdelrahman. 2023. [Arabic Poem Comprehensive Dataset \(APCD\)](#). HuggingFace.

Saad, Motaz. 2019. [Arabic-News](#). GitHub.

Saad, Motaz and Ashour, Mohammed. 2019. [Arabic Billion Words Corpus](#). GitHub. Large-scale Arabic corpus for NLP tasks.

Wenzek, Guillaume and Lachaux, Marie-Anne and Conneau, Alexis and others. 2020. *OPUS Wikipedia: A Large Multilingual Corpus of Wikipedia Articles*. OPUS / NLPL.

Zhang, Biao and Williams, Philip and Titov, Ivan and Sennrich, Rico. 2020. *OPUS-100: A Dataset for Multilingual Machine Translation*. OPUS / NLPL.

A. Dataset Card

Dataset Card for DIA2 Arabic Dataset

Dataset Name:

DIA2: A Comprehensive and Diverse Diacritized Modern Standard Arabic Corpus for Large-Scale NLP Research

Language: *Modern Standard Arabic (MSA) and Classical Arabic (CA)*

Size: 41.9 billion tokens

Key Features:

- Data and Code are publicly available
- Fully automatically diacritized for phonetic and syntactic tasks
- Gold diacritized subset available
- No translated or machine-generated content
- Compiled from credible sources
- Comprehensive cleaning (Preprocessing, URL filtering, deduplication)

Intended Use:

- Language modeling
- Text-to-speech
- MSA/Dialect classification
- Sentence completion
- Diacritics restoration

Limitations:

- No dialectal texts included
- Scarcity of clean gold diacritized data
- No sentence-level deduplication

Citation:

```
@inproceedings{dekmak2026dia2,
  title      = {DIA2: A Comprehensive and Diverse
    Diacritized Modern Standard Arabic Corpus for Large-
    Scale NLP Research},
  author     = {Dekmak et al},
  booktitle  = {Proceedings of the OSACT7 Workshop at
    LREC-COLING 2026},
  year       = {2026}
}
```

B. Deduplication Parameters - Experiment

A mini-experiment was conducted on a sample of 682,004 documents, testing values of 20 and 32 for the number of permutations, and 0.4, 0.5, and 0.8 for the Jaccard similarity threshold. Prior to duplicate identification, 10 known duplicates with specific IDs were injected into the sample, leading to a total of 682,014 documents. This setup allowed us to evaluate the recall of each experiment.

For each deduplication test, we checked how many of these injected duplicates were identified (recall). We also inspected 10 identified duplicates, uniquely identified by each test, to determine if the results were actual duplicates or false positives (precision). The results of the precision and recall tests are presented in Table 7. Precision refers to the proportion of actual duplicates among the identified duplicates. Recall refers to how many of the 10 injected near-duplicates were identified. Given the results in 7, it is noticed that a lower

Permutations	Threshold	Recall	Precision	F1 score
20	0.4	1	0	0
20	0.5	1	0.4	0.57
20	0.8	0.1	1	0.18
32	0.4	1	0	0
32	0.5	1	1	1
32	0.8	0.1	1	0.18

Table 7: Precision and Recall: Precision is the number of actual duplicates identified out of 10 inspected duplicates, and Recall is the number of injected near-duplicates identified out of 10.

threshold (0.4, 0.5) allows more variation between documents before they are flagged as duplicates, which can increase recall (finding more duplicates) but may lead to false positives (incorrectly identifying documents as duplicates). Conversely, a higher threshold (0.8) requires documents to be more similar, which increases precision (correctly identifying true duplicates) but may reduce recall, as some near-duplicates may be missed. We also observed that using 32 permutations consistently provided more accurate and reliable results compared to 20 permutations.

Based on these results, the number of permutations was set to 32 and the Jaccard similarity threshold to 0.5. This balance provided the best trade-off between identifying duplicates and minimizing false positives, given the available computational resources.

CV-18 NER: Augmented Common Voice for Named Entity Recognition from Arabic Speech

Youssef Saidi, Haroun Elleuch, Fethi Bougares

ELYADATA, Paris, France

{youssef.saidi, haroun.elleuch, fethi.bougares}@elyadata.com

Abstract

End-to-end speech Named Entity Recognition (NER) aims to directly extract entities from speech. Prior work has shown that end-to-end (E2E) approaches can outperform cascaded pipelines for English, French, and Chinese, but Arabic remains under-explored due to its morphological complexity, the absence of short vowels, and limited annotated resources. We introduce **CV-18 NER**, the first publicly available dataset for NER from Arabic speech, created by augmenting the Arabic Common Voice 18 corpus with manual NER annotations following the fine-grained Wajood schema (21 entity types). We benchmark both pipeline systems (ASR + text NER) and E2E models based on Whisper and AraBEST-RQ. E2E systems substantially outperform the best pipeline configuration on the test set, reaching 37.0% CoER (AraBEST-RQ 300M) and 38.0% CVER (Whisper-medium). Further analysis shows that Arabic-specific self-supervised pretraining yields strong ASR performance, while multilingual weak supervision transfers more effectively to joint speech-to-entity learning, and that larger models may be harder to adapt in this low-resource setting. Our dataset and models are publicly released, providing the first open benchmark for end-to-end named entity recognition from Arabic speech <https://huggingface.co/datasets/Elyadata/CV18-NER>.

Keywords: Dataset, Arabic Speech, NER, Speech Recognition, Whisper, AraBEST-RQ

1. Introduction

NER from speech is a challenging and under-explored task, particularly for Arabic, a language characterized by morphological richness and linguistic complexity. Traditional approaches typically follow a cascaded two-stage pipeline: (i) transcribe speech to text using an Automatic Speech Recognition (ASR) system, and (ii) apply a NER model to the resulting transcription. However, this pipeline suffers from error propagation, where transcription errors negatively impact downstream entity recognition. To mitigate this issue, recent studies have explored end-to-end (E2E) approaches that predict named entities directly from speech, bypassing intermediate transcripts. Such models have shown promising results in high-resource languages, with successful systems reported for English (Ayache et al., 2025) (Yadav et al., 2020), French (Mdhaftar et al., 2022) (Caubrière et al., 2020) (Szymański et al., 2023), and Chinese (Li et al., 2024) (Chen et al., 2022). These studies demonstrate the viability of joint speech-to-entity learning, and motivate extending this paradigm to morphologically rich languages such as Arabic, which remain under-explored due to the limited availability of annotated datasets and standardized benchmarks. To the best of our knowledge, only one prior study has addressed semantic extraction from Tunisian Arabic speech, which differs from our focus and setting.

In this paper, we introduce **CV-18 NER**, the first publicly available Standard Arabic dataset specifically designed for E2E NER from speech. We leverage the Modern Standard Arabic (MSA) subset of Common Voice 18 (Ardila et al., 2020) and

manually annotate it with named entities following the Wajood (Jarrar et al., 2022) annotation schema. In addition, we propose an E2E Arabic speech NER model based on the Whisper architecture (Radford et al., 2023). Our approach modifies the Whisper sequence-to-sequence framework to jointly transcribe speech and predict named entities. By enriching transcripts with inline BIO-style entity tags and adapting the tokenizer accordingly, the model learns to associate acoustic cues with semantic labels directly.

Our contributions are summarized as follows:

- We present **CV-18 NER**, the first publicly available dataset for NER from Arabic speech, manually annotated with 21 entity types based on the Wajood tagset.
- We develop an end-to-end Whisper-based model that jointly performs speech transcription and NER.
- We compare the performance of Whisper models to the Arabic-specific self-supervised learning (SSL) models AraBEST-RQ (Elleuch et al., 2026) in both cascaded and E2E settings.
- We report experimental results demonstrating that semantic supervision through entity-aware training improves both entity tagging and transcription quality in Whisper-medium, achieving a 1.3 absolute WER point reduction over its ASR counterpart.

2. Dataset

This section introduces the **CV-18 NER** dataset, which is derived from Common Voice 18 (Ardila et al., 2020), an open-source speech corpus consisting of crowd-sourced read speech recorded by a large number of speakers in diverse acoustic conditions. The Arabic subset of Common Voice 18 contains approximately **32 hours** of training data, **12 hours** for development, and **12 hours** for testing. The corpus is released under a CC0 license, allowing unrestricted redistribution of the annotated data, and has been widely used as a benchmark for Arabic ASR systems.

Common Voice was selected as the basis for CV-18 NER for four main reasons. First, its open license makes it possible to publicly release the resulting annotations. Second, its widespread adoption within the speech processing community makes it a familiar and reproducible benchmark. Third, its moderate size makes large-scale manual annotation feasible while still preserving substantial speaker and recording variability. Last, unlike task-specific speech corpora, the diversity and general-domain nature of Common Voice make it particularly suitable for building realistic speech understanding benchmarks.

2.1. Named Entity Schema

For the NER tagset, we conducted a comparative analysis of major Arabic NER corpora, including **AQMAR** (Mohit et al., 2012), **ANERCorp** (Benajiba et al., 2009), and **Wojood** (Jarrar et al., 2022). These corpora differ substantially in tagset granularity. ANERCorp uses a coarse annotation scheme with four entity types (PER, LOC, ORG, MISC), while AQMAR provides a moderately fine-grained annotation with fewer than a dozen categories. In contrast, the **Wojood** schema defines a richer tagset comprising 21 entity types (Jarrar et al., 2022), covering a broader range of semantic categories such as occupations, nationalities, dates, numerical expressions, and websites.

Wojood was therefore selected for CV-18 NER because its fine-grained schema enables more detailed semantic annotation while remaining compatible with both Modern Standard Arabic and dialectal varieties. This level of granularity is particularly suitable for speech understanding tasks, where diverse entity types frequently occur.

2.2. Data annotation

Our data annotation is a pipeline of two sequential stages designed to ensure both annotation efficiency and quality.

1. **Automatic pre-annotation:** In the first stage, several Arabic language models were fine-

tuned and evaluated on the Wojood training set. Among the evaluated models, **AraBERT v2** (Antoun et al., 2020) achieved the best performance and was therefore selected to generate initial pseudo-labeled named entity annotations for the manual transcriptions of all Common Voice 18 recordings. This pre-annotation step significantly reduced the manual annotation effort while providing an initialization aligned with the Wojood annotation guidelines.

We opted for supervised BERT-based models rather than large language model (LLM) annotation because the task requires strict adherence to the Wojood annotation schema. Fine-tuned NER models produce structured and schema-consistent predictions, whereas prompting-based approaches with LLMs often generate inconsistent label formats and require substantial post-processing.

2. **Manual revision:** In the second stage, all pseudo-labeled transcriptions produced during pre-annotation were manually reviewed and corrected

This step was performed by a single annotator who corrected the output of the automatic pre-annotation and ensured its compliance with the Wojood annotation schema. Hence, the inter-annotator agreement score is not reported. We acknowledge that relying on a single annotator can introduce bias and aim to mitigate this limitation when additional annotations can be afforded.

Figure 1 illustrates the annotation workflow used to construct the CV-18 NER dataset, from model-based pre-annotation to manual revision and filtering.

The CV-18 NER annotations follow a BIO tagging scheme consistent with the Wojood annotation guidelines. Each entity span is labeled using beginning (B-) and inside (I-) tags associated with the corresponding entity type. Figure 2 illustrates an example of an Arabic sentence with its tagged representation as well as its English translation.

2.3. CV-18 NER dataset

The Common Voice 18 dataset includes 28,410 utterances for training, 10,471 for development, and 10,471 for testing, totaling 32 hours, 12 hours, and 12 hours of speech, respectively.

Following the annotation pipeline described above and illustrated in Figure 1, utterances that do not contain any Named Entity were removed to focus the experiments on informative segments. After applying this filtering step, the number of utterances in each split was reduced by roughly 75%.

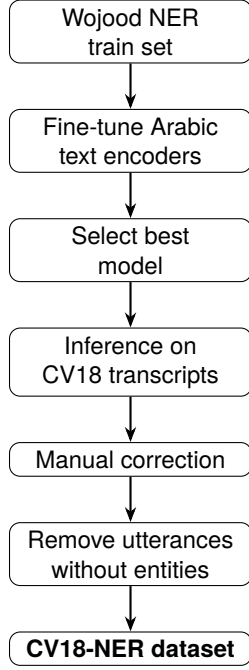


Figure 1: Annotation workflow of CV-18 NER dataset.



Figure 2: Example of enriched transcripts with in-line BIO-style entity markup.

We refer to the resulting subset as **CV-18 NER**. The details of **CV-18 NER** are reported in Table 1.

Split	Train	Dev	Test
# Utterances	7119	2263	2325
Total Duration (h:mm)	8:15	2:54	2:59

Table 1: Statistics of CV-18 NER dataset.

As shown in Table 1, 8h15min of the original 32 training hours remain after filtering out segments without Named Entity annotations. Similarly, 2h54min and 2h59min remain in the development and test sets, respectively. The distribution of Named Entity classes for each split is presented in Table 2.

As we can see from this NER-tags distribution, CV-18 NER dataset exhibits a highly imbalanced class distribution, with the **PERS** category dominating all splits, while several entity types such as **LAW**,

Entity Class	Train	Dev	Test
CARDINAL	311	139	138
CURR	2	0	0
DATE	1176	516	597
EVENT	112	26	50
FAC	112	46	58
GPE	541	327	288
LANGUAGE	237	108	101
LAW	1	3	2
LOC	201	75	97
MONEY	51	29	29
NORP	1280	458	378
OCC	588	167	249
ORDINAL	289	103	87
ORG	211	49	117
PERCENT	2	2	7
PERS	5314	1250	1373
PRODUCT	40	2	3
QUANTITY	28	7	17
TIME	539	242	243
UNIT	0	0	0
WEBSITE	15	6	8

Table 2: Distribution of NER-tags in the CV-18 NER dataset.

CURR, **PERCENT**, and **PRODUCT** are only sparsely represented. This imbalance can be explained by the combination of adopting the fine-grained Wojoood tagset, which provides broad coverage across multiple semantic domains, and the nature of Common Voice prompts, which consist of short crowd-sourced read sentences that are not specifically curated for named-entity richness. As a result, frequently occurring entity types such as **PERS** and **GPE** are well represented, whereas rarer semantic categories appear only occasionally. Although this imbalance poses challenges for learning low-frequency classes, it reflects realistic conditions in speech applications and Named Entity Recognition tasks, where entity occurrences typically follow long-tailed distributions (Nemoto et al., 2025; de Alvis and Seneviratne, 2024; Zevallos et al., 2023). The resulting dataset therefore provides a challenging and representative benchmark for evaluating end-to-end speech NER systems under realistic and limited-resource conditions.

3. Speech NER systems

To assess the effectiveness of directly predicting named entities from Arabic speech, we implemented and compared two system architectures: (i) a pipeline approach of Automatic Speech Recognition (ASR) followed by a NER system applied to the ASR outputs, and (ii) an end-to-end approach that jointly performs transcription and named entity prediction.

3.1. Model families

We evaluated two complementary families of speech representation models. The first family consists of Whisper models (Radford et al., 2023), which are weakly supervised multilingual models trained on large-scale speech-text data and widely used as strong baselines in speech processing tasks. We used the **Whisper-medium** and **Whisper-large-v3** pretrained models.

The second family consists of AraBEST-RQ models (Elleuch et al., 2026), a self-supervised learning (SSL) model based on the BEST-RQ framework. Unlike Whisper, AraBEST-RQ models are trained primarily on Arabic audio using self-supervised objectives and do not rely on paired speech-text supervision. We evaluated the 300M and 600M parameter variants.

The AraBEST-RQ 300M model is pretrained on approximately 6k hours of crawled Arabic speech. The 600M model is available in two variants: one trained on the same 6k-hour corpus and another trained on approximately 14k hours of Arabic speech obtained by combining crawled data with publicly available datasets, including Common Voice 16.

In addition to speech models, we evaluated four text-based NER systems namely **AraBERT v2**, **AraBERTv0.2**, **CAMELbert-MSA**, and **CAMELbert-MIX**.

Comparing Whisper and AraBEST-RQ allows us to evaluate NER-from-speech performance across different representation paradigms: large-scale weakly supervised multilingual models on the one hand, and language-specific self-supervised representations on the other. We believe that such a comparison provides meaningful baselines for future research on Arabic speech understanding.

3.2. Pipeline Speech NER

The pipeline system follows the classical two-stage approach to speech NER. In the first stage, an ASR module transcribes the input speech signal into text. In the second stage, a text-based NER model assigns named entity labels to the resulting transcription.

ASR systems are implemented using the Whisper and AraBEST-RQ models described in Section 3.1. The generated transcriptions are then processed by the text-based NER models to produce the final entity annotations.

Fine-tuning was performed using the CV-18 NER training set as described in Table 1. This pipeline configuration serves as a baseline for comparison with the end-to-end approach described below.

3.3. End-to-End Speech NER

In contrast to the pipeline approach, end-to-end (E2E) speech NER systems directly predict named entity annotations from raw speech without relying on an intermediate transcription stage. This formulation enables the joint learning of acoustic and semantic representations and reduces the error propagation typically observed in pipeline systems.

Our end-to-end models are implemented using Whisper and AraBEST-RQ architectures described in Section 3.1, which are fine-tuned to generate entity-aware transcriptions. Entity-aware training is performed by modifying the reference transcripts to include inline entity tags using BIO-style markup tokens, as described in Section 2.2. The models are trained to generate these enriched transcripts directly from speech, allowing them to learn associations between acoustic patterns and semantic labels while preserving the transcription task.

To support structured entity prediction, the tokenizer vocabularies are extended with entity markers treated as special tokens (e.g., <B-PERS>, <I-ORG>), ensuring that entity tags are handled as atomic units during encoding and decoding.

4. Experiments and Results

4.1. Experimental Setup

The text encoder models were fine-tuned on a single NVIDIA Tesla P100 GPU with 16GB of memory. All experiments were conducted using a maximum sequence length of 512 tokens and a batch size of 64. We used the AdamW optimizer with a learning rate of 2×10^{-5} , $\epsilon = 10^{-8}$, and weight decay of 0.01. The NER model was trained using CrossEntropyLoss.

For speech experiments, audio signals were resampled to 16 kHz and the reference transcriptions were normalized following (Chowdhury et al., 2021). Specifically, we (a) removed punctuation, (b) removed diacritics, Hamzas, and Maddas, and (c) transliterated Eastern Arabic numerals into Western Arabic numerals. All speech experiments were conducted using the SpeechBrain toolkit (Ravanelli et al., 2024). Models were trained on an NVIDIA V100 GPU with 32 GB of VRAM, except for Whisper-large-v3 which was trained on an A100 GPU with 80 GB of VRAM.

AraBEST-RQ models use the same architecture and training hyperparameters for both ASR and E2E NER tasks. Encoder representations are passed through a feedforward classification head followed by a linear projection and softmax layer.

Training uses the CTC loss with the AdamW optimizer, with learning rates of 3×10^{-4} for newly initialized layers and 5×10^{-5} for the pretrained encoder. Training is performed with a batch size of 8 and gradient accumulation over 2 steps. BEST-RQ models use a character-level tokenizer. The ASR systems use a vocabulary of 41 symbols, while the NER systems use an extended vocabulary of 79 symbols including entity tags. Each entity tag is treated as a single atomic token.

Whisper-medium and Whisper-large-v3 models are fine-tuned end-to-end using the negative log-likelihood loss and the AdamW optimizer with a learning rate of 1×10^{-5} and weight decay of 0.01. Training is performed with a batch size of 4 and gradient accumulation over 4 steps. For E2E NER experiments, entity tags are added as special tokens to the Whisper tokenizer and the embedding matrix is resized accordingly.

Text-based NER models are evaluated using micro-averaged F1-score. Since text-based systems operate on fixed input sentences, predicted labels can be directly aligned with reference labels, making F1-score an appropriate evaluation metric.

For speech-based systems, transcription errors and alignment differences make token-level comparison unreliable. Therefore, speech NER systems are evaluated using error-rate metrics derived from speech recognition and spoken language understanding evaluation. Transcription quality is measured using Word Error Rate (WER), computed after removing special entity tokens from the model outputs. Named entity recognition performance is evaluated using Concept Error Rate (CoER) and Concept-Value Error Rate (CVER), which are adapted from spoken language understanding evaluation. WER, CoER, and CVER follow the standard error-rate formulation:

$$\text{Error Rate} = \frac{S + D + I}{N},$$

where S , D , and I denote substitutions, deletions, and insertions, and N is the number of reference units.

CoER evaluates the correctness of entity labels only (e.g., B-PERS, I-ORG), independently of the lexical content. CVER evaluates both entity type and entity span by comparing tag-value pairs extracted from the annotated transcripts. Lower values indicate better performance.

4.2. Pipeline Speech NER results

In this section, we present the performance of the pipeline Speech NER system. We used Whisper in zero-shot (ZS) and fine-tuning (FT) settings for the speech-to-text stage. AraBEST-RQ was used only under the fine-tuning (FT) setting. For the text-based NER stage, we evaluated the four different models mentioned in 3.1. Before evaluating the full pipeline (ASR+NER), we first assess each component individually.

4.2.1. Speech Recognition Results

We compared AraBEST-RQ to Whisper-medium and Whisper-large-v3 in both zero-shot and fine-tuned settings. Fine-tuning was performed on the filtered training subset (8 hours 15 minutes, Table 1). Table 3 reports the WER obtained by all ASR models on the validation and test splits.

The results show that zero-shot Whisper performs very poorly on CV-18 NER, with WER above 92% for Whisper-large-v3 and 107% for Whisper-medium. Fine-tuning substantially improves performance, especially for Whisper-medium, which achieves the best WER among Whisper models with **14.9%** on the validation set and **22.0%** on the test set. In contrast, Whisper-large-v3 shows a smaller improvement after fine-tuning, reaching 31.8% on validation and 32.3% on test, and remains significantly worse than the medium model. Preliminary analysis of its outputs indicates that Whisper-large-v3 sometimes produces English translations instead of Arabic transcriptions or transliterations into Latin script, which warrants further investigation. Such behaviour is, however, known and has been previously observed and reported in works such as (Talafha et al., 2024).

Among AraBEST-RQ models, the 300M 6k variant achieves the lowest validation WER (**11.2%**) and the lowest test WER (**15.1%**), outperforming all Whisper models. Other AraBEST-RQ variants show slightly higher WERs, with the 600M 6k and 600M 14k models achieving up to 17.4% on the test set.

Overall, these results highlight that fine-tuning is critical for Whisper-based ASR on limited Arabic data and that increasing model size alone does not guarantee better performance. AraBEST-RQ demonstrates a more consistent improvement across both validation and test sets, making it the preferred choice in this low-resource scenario.

4.2.2. Text-based NER Results

We fine-tune several Arabic BERT-based (Devlin et al., 2019) models on the CV-18 NER train set transcriptions and evaluate them using **micro-F1** on the transcriptions of the validation and test sets

Model	Valid.	Test
Whisper-medium (ZS)	107.4	112.3
Whisper-medium (FT)	14.9	22.0
Whisper-large-v3 (ZS)	92.7	95.1
Whisper-large-v3 (FT)	31.8	32.3
AraBEST-RQ 300M 6k (FT)	11.2	15.1
AraBEST-RQ 600M 6k (FT)	12.2	17.4
AraBEST-RQ 600M 14k (FT)	11.2	16.4

Table 3: Validation and test Word Error Rates (WER %) of ASR models on the CV-18 NER dataset. FT: Fine-tuning. ZS: Zero-shot. Best scores per metric are reported in bold.

of the same dataset. The top performing models were then used for pipeline systems.

Model	Validation	Test
CamelBERT-Mix	79.66	77.24
CamelBERT-MSA	79.76	76.13
AraBERTv2	81.06	76.58
AraBERTv0.2	81.10	77.34

Table 4: Validation and test micro-F1 scores (%) of BERT-based NER models on the CV-18 NER dataset. Best scores per column are reported in bold.

4.2.3. Pipeline Systems Results

Based on the ASR and text NER results reported above, we evaluated the Speech NER pipeline system using the five fine-tuned ASR systems, followed by the four chosen BERT-based systems to perform NER on the generated transcriptions. The obtained results are presented in Table 5.

While transcription quality generally influences downstream performance, the results show that lower WER does not necessarily lead to better entity extraction. In particular, Whisper-medium pipelines outperform AraBEST-based pipelines despite higher transcription error rates.

On the CV-18 NER test split, the lowest CoER is obtained when AraBERTv0.2 is applied to the output of the fine-tuned Whisper-medium ASR system, achieving 51.3 CoER and 50.2 CVER. These results suggest that while ASR quality influences downstream entity extraction, it is not the only factor; the choice of ASR and NER models (and the nature of ASR errors) substantially impacts the final pipeline performance.

A qualitative analysis shows that the best pipeline system performs well on high-frequency entity

NER Model	ASR Model	CoER	CVER
CamelBERT-Mix	AraBEST-RQ 600M 14k	66.4	65.8
	AraBEST-RQ 300M 6k	66.1	65.4
	AraBEST-RQ 600M 6k	67.7	67.0
	Whisper-large-v3 (FT)	57.7	56.7
	Whisper-medium (FT)	57.2	55.9
CamelBERT-MSA	AraBEST-RQ 600M 14k	66.3	65.7
	AraBEST-RQ 300M 6k	66.3	65.7
	AraBEST-RQ 600M 6k	67.9	67.2
	Whisper-large-v3 (FT)	57.8	56.7
	Whisper-medium (FT)	57.4	56.1
AraBERTv2	AraBEST-RQ 600M 14k	72.0	71.3
	AraBEST-RQ 300M 6k	71.5	70.9
	AraBEST-RQ 600M 6k	72.0	71.4
	Whisper-large-v3 (FT)	63.6	62.5
	Whisper-medium (FT)	64.5	63.0
AraBERTv0.2*	AraBEST-RQ 600M 14k	63.1	62.6
	AraBEST-RQ 300M 6k	62.7	62.0
	AraBEST-RQ 600M 6k	63.8	63.2
	Whisper-large-v3 (FT)	54.1	53.5
	Whisper-medium (FT)*	51.3	50.2

Table 5: Pipeline evaluation on CV-18 NER test set with grouped NER models. Columns show CoER and CVER for each ASR + NER combination. Best scores per metric are reported in bold. FT: Fine-tuning. The best configurations are followed by *.

types such as **LANGUAGE** and **GPE**, while performance drops sharply for low-frequency categories such as **PRODUCT**, **LAW**, and **QUANTITY**. We also noted that ASR errors, particularly those stemming from phonetic ambiguities in MSA, contribute significantly to the degradation of NER performance.

4.3. End-to-end Speech NER results

This section presents the performance of our E2E Speech NER system, which is based on Whisper and AraBEST-RQ models fine-tuned in an E2E fashion for entity extraction.

Table 6 shows that the AraBEST-RQ 600M models failed to produce correct outputs in E2E fine-tuning, leading to near-random predictions. However, AraBEST-RQ 300M outperforms all other systems in WER and CoER. Whisper-medium is, however, the best in terms of CVER. Whisper-large-v3 is also largely outperformed by the smaller Whisper-medium and AraBEST-RQ 300M.

Model	WER	CoER	CVER
AraBEST-RQ 300M 6k	16.0	37.0	43.5
AraBEST-RQ 600M 6k	99.4	87.1	94.0
AraBEST-RQ 600M 14k	99.6	99.8	100
Whisper-medium	20.7	38.8	38.0
Whisper-large-v3	41.9	59.6	59.1

Table 6: E2E NER from speech results on the CV-18 NER test set. Best scores per metric are reported in bold. Entity tags are removed before WER computation. The same normalization is applied to all systems.

We did the same entity-wise investigation, and the results show improved performance on several core entity types. Notably, the E2E model achieves better results for **LANGUAGE** and **ORDINAL**, along with significant gains on **PERS** and **GPE** compared to the pipeline. However, performance remains limited on low-frequency types such as **LAW**, **PERCENT**, and **PRODUCT**, which were completely missed or poorly predicted. These results demonstrate the potential of end-to-end training to capture richer acoustic-entity correlations, while also highlighting ongoing challenges in recognizing rare entity types from speech. A detailed breakdown of errors per entity for the two best-performing models, Whisper-medium and AraBEST-RQ 300M 6k, can be found in Tables 7 and 8 in the Appendix.

5. Discussion and Analysis

This section provides a detailed analysis of the experimental findings across both pipeline and end-to-end (E2E) configurations, with particular emphasis on representation paradigms, error propagation, and entity-type behavior.

5.1. ASR Quality vs. Downstream Entity Recognition

Despite AraBEST-RQ 300M achieving substantially lower WER than Whisper-medium, Whisper-based pipelines consistently obtain lower CoER and CVER scores. This suggests that transcription quality alone does not fully determine entity extraction performance.

These findings confirm that classical pipeline systems remain highly sensitive to transcription errors, particularly for entity spans where even a single token substitution may invalidate the entire tag-value pair, resulting in high CVER.

5.2. Weakly Supervised vs. Self-Supervised Representations

A central contribution of this work is the comparison between Whisper (weakly supervised multilingual training) and AraBEST-RQ (Arabic-specific self-supervised learning).

In the pipeline setting, AraBEST-RQ 300M performs strongly as an ASR model, demonstrating that Arabic-specific SSL representations are competitive when fine-tuned for transcription, with all AraBEST-RQ models outperforming Whisper models. However, this pattern shifts in E2E NER from speech. While AraBEST-RQ 300M is still the best-performing model in both WER and CoER, it falls behind Whisper-medium by 5.5 absolute points in CVER. It is worth noting that AraBEST-RQ 300M is

less than half the size of Whisper-medium (769M parameters) in terms of the number of parameters. This contrast suggests that:

- SSL-based encoders trained without paired speech-text supervision adapt well to ASR through CTC fine-tuning,
- but larger SSL-based models struggle to converge in joint transcription and semantic tagging, as observed for the AraBEST-RQ 600M variants.

We also observe that well-tuned smaller-scale models can provide strong alignment between acoustic and lexical-semantic representations, which is crucial for direct entity prediction. Interestingly, Whisper-large-v3 does not outperform Whisper-medium in E2E training. This may be attributed to over-parameterization relative to the limited 8-hour training subset, leading to suboptimal adaptation under low-resource fine-tuning conditions. This finding aligns with recent observations in Arabic speech processing (Elleuch et al., 2025), where Whisper-medium similarly outperformed Whisper-large-v3 when fine-tuned on dialect-specific data of a similar scale (16 hours) for ASR tasks.

5.3. End-to-End vs. Pipeline: Error Propagation Revisited

Comparing the best pipeline configuration (AraBERTv0.2 + Whisper-medium) with the best E2E results, we observe:

- Lower CoER in E2E (37.0 vs. 51.3),
- Lower CVER in E2E (38.0 vs. 50.2).

These improvements confirm that eliminating the intermediate transcription stage reduces error propagation. In pipeline systems, entity prediction depends strictly on lexical correctness. Any ASR substitution inside an entity span typically results in a CVER penalty.

In contrast, the E2E model can partially recover entity type information from acoustic cues even when lexical realization is imperfect. This indicates that semantic supervision encourages the model to encode entity-relevant acoustic patterns, rather than relying exclusively on textual reconstruction. Additionally, E2E training appears to act as a form of task-aware regularization, improving semantic robustness even when transcription accuracy (WER) is not the lowest among models.

5.4. Entity-Type Behavior

The dataset has an imbalanced distribution, with `PERS`, `GPE`, and `DATE` dominating the corpus, while categories such as `LAW`, `PRODUCT`, and `PERCENT` are extremely sparse.

Both pipeline and E2E systems show:

- Strong performance on high-frequency classes (`PERS`, `GPE`, `LANGUAGE`),
- Severe degradation on rare classes (`LAW`, `PRODUCT`, `QUANTITY`).

However, the E2E model demonstrates improved robustness on certain mid-frequency categories such as `ORDINAL` and `TIME`, suggesting that joint acoustic-semantic training may help disambiguate numeric and temporal expressions that are phonetically structured.

Rare classes remain challenging for both paradigms. Given the extremely small number of training examples (e.g., `LAW` and `CURR`), the models are unable to learn stable acoustic-semantic mappings.

5.5. Key Takeaways

From these experiments, we draw several conclusions:

1. End-to-end training substantially reduces entity-level error compared to strong pipeline baselines.
2. Language-specific SSL representations can be competitive for E2E NER from speech with a 300M parameters model outperforming Whisper-medium in both WER and CoER despite being less than half its size.
3. Smaller models show better performance in ASR and E2E NER from speech than their larger counterparts.
4. Pipeline systems remain highly sensitive to ASR errors, especially at the entity-span level (CVER).
5. Entity imbalance remains the primary bottleneck for Arabic speech NER.

Overall, the results demonstrate that E2E speech NER is not only viable for Modern Standard Arabic but can significantly outperform cascade approaches when appropriate pretrained representations are used.

6. Conclusion

In this paper, we introduced **CV-18 NER**, the first annotated Arabic dataset for Speech Named Entity Recognition. Initial experiments with SoTA models demonstrate that E2E speech NER is viable for Arabic and significantly outperforms cascade approaches when appropriate pretrained representations are used. We believe that this resource, along with the baseline results provided in this paper, will foster the development of better end-to-end speech NER for the Arabic language and its dialects.

7. Bibliographical References

- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resources Association.
- Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. [Common voice: A massively-multilingual speech corpus](#). In *LREC*, pages 4218–4222, Marseille, France. European Language Resources Association.
- Gil Ayache, Menachem Pirchi, Aviv Navon, Aviv Shamsian, Gill Hetz, and Joseph Keshet. 2025. [Whisperer: Unified open named entity and speech recognition](#).
- Yassine Benajiba, Paolo Rosso, and José-Miguel Benedí. 2009. [Anersys: An arabic named entity recognition system based on maximum entropy](#). In *Conference on Intelligent Text Processing and Computational Linguistics*.
- Antoine Caubrière, Sophie Rosset, Yannick Estève, Antoine Laurent, and Emmanuel Morin. 2020. [Where are we in named entity recognition from speech?](#) In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4514–4520, Marseille, France. European Language Resources Association.
- Boli Chen, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Meishan Zhang, and Fei Huang. 2022. [Aishell-ner: Named entity recognition from chinese speech](#). In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8352–8356.

- Shammur Absar Chowdhury, Amir Hussein, Ahmed Abdelali, and Ahmed Ali. 2021. [Towards One Model to Rule All: Multilingual Strategy for Dialectal Code-Switching Arabic ASR](#). In *Interspeech 2021*, pages 2466–2470.
- Charika de Alvis and Suranga Seneviratne. 2024. [A survey of deep long-tail classification advancements](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Haroun Elleuch, Youssef Saidi, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2025. [Elyadata & lia at nadi 2025: Asr and adi subtasks](#).
- Haroun Elleuch, Ryan Whetten, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2026. [Ara-best-rq: Multidialectal arabic ssl](#). In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Accepted for publication.
- Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. [Wojoood: Nested Arabic named entity corpus and recognition using BERT](#). In *LREC*, pages 3626–3636, Marseille, France. ELRA.
- Yuang Li, Jiawei Yu, Min Zhang, Mengxin Ren, Yanqing Zhao, Xiaofeng Zhao, Shimin Tao, Jinsong Su, and Hao Yang. 2024. [Using Large Language Model for End-to-End Chinese ASR and NER](#). In *Interspeech 2024*, pages 822–826.
- Salima Mdhaffar, Jarod Duret, Titouan Parcollet, and Yannick Estève. 2022. [End-to-end model for named entity recognition from speech without paired training data](#). In *Interspeech 2022*, pages 4068–4072.
- Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A. Smith. 2012. [Recall-oriented learning of named entities in Arabic Wikipedia](#). In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 162–173, Avignon, France. Association for Computational Linguistics.
- Sota Nemoto, Shunsuke Kitada, and Hitoshi Iyatomi. 2025. [Majority or minority: Data imbalance learning method for named entity recognition](#). *IEEE Access*, 13:9902–9909.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. [Robust speech recognition via large-scale weak supervision](#). In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.
- Mirco Ravanelli, Titouan Parcollet, Adel Moumen, Sylvain De Langen, Cem Subakan, Peter Plantinga, Yingzhi Wang, Pooneh Mousavi, Luca Della Libera, Artem Ploujnikov, Francesco Paissan, Davide Borra, Salah Zaiem, Zeyu Zhao, Shucong Zhang, Georgios Karakasidis, Sung-Lin Yeh, Pierre Champion, Aku Rouhe, Rudolf Braun, Florian Mai, Juan Zuluaga-Gomez, Seyed Mahed Mousavi, Andreas Nautsch, Ha Nguyen, Xuechen Liu, Sangeet Sagar, Jarod Duret, Salima Mdhaffar, Gaëlle Laperrière, Mickael Rouvier, Renato De Mori, and Yannick Estève. 2024. [Open-source conversational ai with speechbrain 1.0](#). *J. Mach. Learn. Res.*, 25(1).
- Piotr Szymański, Lukasz Augustyniak, Mikolaj Morzy, Adrian Szymczak, Krzysztof Surdyk, and Piotr Żelasko. 2023. [Why aren't we NER yet? artifacts of ASR errors in named entity recognition in spontaneous speech transcripts](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1746–1761, Toronto, Canada. Association for Computational Linguistics.
- Bashar Talafha, Karima Kadaoui, Samar Mohamed Magdy, Mariem Habiboullah, Chafei Mohamed Chafei, Ahmed Oumar El-Shangiti, Hiba Zayed, Mohamedou cheikh tourad, Rahaf Alhamouri, Rwa Assi, Aisha Alraeesi, Hour Mohamed, Fakhraddin Alwajih, Abdelrahman Mohamed, Abdellah El Mekki, El Moatez Billah Nagoudi, Benelhadj Djelloul Mama Saadia, Hamzah A. Alsayadi, Walid Al-Dhabyani, Sara Shatnawi, Yasir Ech-Chammakhy, Amal Makouar, Yousra Berrachedi, Mustafa Jarrar, Shady Shehata, Ismail Berrada, and Muhammad Abdul-Mageed. 2024. [Casablanca: Data and models for multidialectal arabic speech recognition](#).
- Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah. 2020. [End-to-end named entity recognition from english speech](#).
- Rodolfo Zevallos, Mireia Farrús, and Núria Bel. 2023. [Frequency balanced datasets lead to better language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7859–7872, Singapore. Association for Computational Linguistics.

Appendix

A. Per-Entity Error Statistics for the Best-Performing E2E Systems

Tables 7 and 8 present a breakdown by entity of error types for the two best-performing end-to-end NER-from-speech systems : Whisper-medium and AraBEST-RQ 300M 6k. For each entity tag, we report the number of insertions (INS), deletions (DEL), and substitutions (SUB), as well as the Concept Error Rate (CoER, %).

Table 7: Per-entity CoER results for Whisper-medium (E2E configuration). INS: insertions, DEL: deletions, SUB: substitutions.

Tag	INS	DEL	SUB	CoER (%)
B-CARDINAL	2	10	31	36.8
I-CARDINAL	1	4	9	93.3
B-DATE	29	43	42	31.2
I-DATE	11	59	46	50.4
B-EVENT	4	14	10	112.0
I-EVENT	4	8	14	108.3
B-FAC	1	19	23	95.6
I-FAC	1	2	8	100.0
B-GPE	11	27	74	42.3
I-GPE	2	5	11	81.8
B-LANGUAGE	0	7	4	10.9
B-LAW	0	1	0	100.0
B-LOC	16	26	23	108.3
I-LOC	0	14	23	100.0
B-MONEY	0	5	5	76.9
I-MONEY	1	3	8	75.0
B-NORP	16	72	53	49.0
I-NORP	7	30	33	77.8
B-OCC	10	42	45	47.1
I-OCC	6	21	18	104.7
B-ORDINAL	2	7	24	38.4
I-ORDINAL	0	0	1	100.0
B-ORG	3	27	14	68.8
I-ORG	1	16	24	91.1
B-PERCENT	0	1	1	100.0
I-PERCENT	0	0	5	100.0
B-PERS	40	125	88	23.1
I-PERS	29	61	32	45.9
B-PRODUCT	0	1	2	100.0
I-PRODUCT	0	0	0	0.0
B-QUANTITY	1	3	6	111.1
I-QUANTITY	0	1	7	100.0
B-TIME	10	26	23	39.9
I-TIME	7	20	25	54.7
B-WEBSITE	0	0	7	100.0
I-WEBSITE	0	0	0	0.0

Table 8: Per-entity CoER results for AraBEST-RQ 300M 6k (E2E configuration). INS: insertions, DEL: deletions, SUB: substitutions.

Tag	INS	DEL	SUB	CoER (%)
B-CARDINAL	4	3	26	28.2
I-CARDINAL	0	5	3	53.3
B-DATE	34	31	41	29.0
I-DATE	17	43	38	42.6
B-EVENT	8	6	7	84.0
I-EVENT	5	9	7	84.0
B-FAC	3	21	11	77.8
I-FAC	2	4	8	107.7
B-GPE	13	24	45	30.8
I-GPE	2	3	7	54.5
B-LANGUAGE	1	3	8	11.9
B-LAW	0	1	0	100.0
B-LOC	11	18	14	71.7
I-LOC	2	8	14	64.9
B-MONEY	0	6	0	46.2
I-MONEY	1	0	8	56.3
B-NORP	25	57	52	46.5
I-NORP	7	31	29	74.4
B-OCC	7	38	23	33.0
I-OCC	5	9	14	65.1
B-ORDINAL	10	6	15	36.0
I-ORDINAL	0	0	0	0.0
B-ORG	5	17	7	44.6
I-ORG	4	12	12	56.0
B-PERCENT	0	0	2	100.0
I-PERCENT	0	0	5	100.0
B-PERS	70	86	39	17.7
I-PERS	37	52	29	43.9
B-PRODUCT	0	1	2	100.0
I-PRODUCT	0	0	0	0.0
B-QUANTITY	1	1	3	55.6
I-QUANTITY	4	2	1	87.5
B-TIME	7	12	21	27.0
I-TIME	6	22	11	41.1
B-WEBSITE	1	2	2	62.5
I-WEBSITE	0	0	0	0.0

ARHAHA 2026: The Shared Task on Arabic Humor Automatic Generation

Ameera Almasoud, Hend Al-Khalifa, Reem Alqifari, Nourah Alangari and Manal Albahlal

College of Computer and Information Sciences, King Saud University
Riyadh, Saudi Arabia
ammalmasoud|hendk|ralqifari|nmalangari|albahlal@ksu.edu.sa

Abstract

Humor generation remains one of the most challenging tasks in natural language processing, particularly in Arabic, where cultural context, dialectal variation, and linguistic nuances are central to comedic effect. In this paper, we present the ARHAHA 2026 shared task on constrained Arabic humor generation. The task requires systems to generate jokes that incorporate a given pair of words while adhering to safety and cultural constraints. We describe the task design, dataset construction, and evaluation framework, which combines automatic validation with human evaluation. Nine teams registered for the shared task; among them, three submitted final system outputs and two provided system description papers. Each participating system generated 1,200 Arabic jokes. For each system, a subset of 300 jokes was selected for evaluation by three independent annotators. The evaluation considered humor quality, originality, lexical constraint compliance, and safety. The results show that participating systems can produce safe and original content. However, generating genuinely humorous outputs remains difficult. The top-performing system was judged humorous in only 5.01% of outputs, highlighting the inherent difficulty of computational humor generation. All three systems maintained very low rates of policy violations and stereotyping, demonstrating the effectiveness of constrained generation for safe content production. However, the very low humor rates indicate a substantial gap between generating fluent, constraint-compliant text and producing genuinely funny content. The top-performing system achieves stronger performance across originality, lexical compliance, and safety, resulting in a final score of 49.25, compared to 44.62 for the second-ranked system and 35.99 for the third-ranked system. These results reveal that humor generation, rather than safety or constraint adherence, is the dominant bottleneck in constrained Arabic humor generation.

Keywords: Arabic humor generation, shared task, constrained text generation, computational humor, natural language generation

1. Introduction

Humor is one of the most complex forms of human communication. Successful humor often relies on unexpected semantic relationships, cultural references, pragmatic interpretation, and narrative timing (Raskin, 1985). For computational systems, generating humor is significantly more difficult than generating ordinary text because humor requires both creativity and an understanding of incongruity (Mihalcea and Strapparava, 2005). Recent advances in large language models have enabled impressive progress in natural language generation tasks. However, humor generation remains a challenging problem. Models often produce fluent sentences but fail to deliver a genuine humorous effect. This difficulty arises because humor often depends on subtle narrative structures, unexpected reinterpretations, and culturally grounded references.

Most previous computational humor research has focused on humor detection, sarcasm classification, or irony detection. More recently, shared tasks such as MWAHAHA (Castro et al., 2026) have begun to address humor generation directly. In contrast, humor generation has received relatively little attention, particularly for languages other than English. Arabic humor generation remains largely unexplored despite the rich cultural traditions of humor in Arabic-speaking communities.

To address this gap, the ARHAHA shared task was introduced to encourage research on Arabic humor generation. Participants are required to generate humorous Arabic sentences given a pair of words that must appear in the text. These lexical constraints ensure that systems must generate novel humorous content rather than retrieving memorized jokes from training data.

The main contributions of the ARHAHA shared task are: (1) introducing the first benchmark dataset for constrained Arabic humor generation, (2) establishing a hybrid evaluation protocol combining automatic validation and human evaluation, and (3) providing insights into different computational approaches for humor generation in Arabic.

This overview paper describes the design and organization of the ARHAHA shared tasks. This paper presents the task description, dataset construction, and evaluation methodology, followed by the participating systems, results, and discussion. It concludes with key insights and directions for future research.

2. Task Description

The ARHAHA shared task focuses on constrained Arabic humor generation. In this task, systems are provided with a pair of Arabic words and must generate a short humorous text that includes both words. The generated joke must respect a maximum length constraint of 100 characters and

integrate the two words naturally within the humorous narrative.

Unlike open-ended humor generation tasks, ARHAHA introduces a set of explicit constraints designed to encourage originality, safety, and cultural appropriateness. These constraints ensure that participating systems generate novel humorous content rather than retrieving existing jokes from online sources or memorized datasets.

The goal of the task is to develop systems capable of producing genuinely humorous Arabic text while respecting lexical constraints and ethical guidelines. By imposing controlled constraints, the task encourages creative generation and prevents models from reproducing widely circulating jokes found on the web.

2.1 Humor Generation Constraints

To ensure safe, culturally appropriate, and meaningful humor generation, all submissions must comply with three main constraints.

- **Content Safety:** Generated jokes must avoid harmful or offensive content, including insults, hate speech, discrimination, sexual or explicit references, and personal attacks. Humor targeting individuals, groups, or cultures in a harmful or demeaning manner is strictly prohibited.
- **Benign Humor:** The task permits lighthearted and non-harmful humor, such as exaggerated situations, fictional or animal-based scenarios, and common Arabic joke archetypes. Dialect-based misunderstandings are allowed when they arise from harmless linguistic differences and do not demean any group. This aligns with the concept of benign violation, where humor is surprising yet non-offensive.
- **Originality:** All generated jokes must be original and not reproduced from existing sources. Systems must avoid copying or rephrasing widely known jokes, memes, or online content.

These constraints were designed to encourage creative, culturally appropriate, and responsible humor generation.

3. Dataset Construction

To evaluate constrained humor generation, we constructed a dataset consisting of 1,200 Arabic word pairs, each serving as a prompt for generating a humorous text. For each prompt, participating systems receive two Arabic words and must generate a short joke that integrates both words naturally within the narrative. Unlike traditional humor datasets that rely on existing jokes, our dataset was designed to encourage creative generation rather than retrieval. Instead of providing complete jokes, the dataset provides word-pair prompts that require systems to construct new humorous situations. The word pairs were manually curated by the organizers to encourage humorous incongruity and diverse

narrative scenarios. The selection process aimed to balance semantic distance and contextual plausibility, ensuring that systems must construct creative connections between the two words.

3.1 Word Pair Design

The evaluation dataset consists of 1,200 manually curated Arabic word pairs evenly distributed across ten semantic categories, with 120 pairs per category. These categories were designed to cover a wide spectrum of humor-inducing scenarios, including logical contradiction, illogical context, role reversal, scale or size violations, impossible behaviors, and misuse of objects or concepts. In addition, the dataset includes everyday domains such as social life, workplace interactions, technology and digital culture, and the animal world, where anthropomorphic or unexpected behaviors frequently give rise to humor. This structured categorization ensures diversity in the evaluation set and enables a more comprehensive assessment of systems' ability to generate humor across different semantic and contextual dimensions.

3.2 Prompt Format

Each dataset instance contains three fields:

- **pair ID:** a unique identifier for the prompt
- **word1:** the first required word
- **word2:** the second required word

Given this pair, systems must generate a humorous Arabic text that includes both words while respecting the task constraints.

4. Competition Phases

The ARHAHA shared task was organized into two main phases to support system development and ensure fair evaluation. These phases were implemented through the Codabench platform and allowed participants to progressively test and finalize their systems.

4.1 Development Phase

The development phase took place from December 8, 2025 to February 9, 2026. During this stage, participants were provided with the task dataset and were allowed to build, fine-tune, and test their humor generation systems. Participants could submit multiple trial runs through the Codabench platform. Each submission was automatically processed by the evaluation pipeline, which performed format validation checks to ensure that the submitted outputs satisfied the structural requirements of the task.

The primary objective of this phase was to help participants verify that their systems generated outputs in the correct format and successfully satisfied the task constraints, including the required word inclusion and length limitations. Importantly, humor quality was not evaluated during this phase, and the leaderboard reflected only compliance with submission requirements.

This stage allowed teams to iteratively refine their models and confirm that their systems were functioning correctly before submitting their final runs in the official evaluation phase.

4.2 Final Evaluation Phase

The final evaluation phase was conducted from February 10, 2026 to February 17, 2026. During this stage, participating teams were required to submit their final system outputs through the Codabench platform. Each submission first underwent automatic validation to ensure compliance with the task constraints and submission format. After passing the automatic checks, the generated jokes were then evaluated through a human evaluation process to assess humor quality, originality, and adherence to safety guidelines.

Unlike the development phase, this stage determined the official ranking of participating systems. The final leaderboard was established based on the evaluation results obtained from the human judging process.

5. Evaluation Methodology

Because humor quality is inherently subjective and cannot be reduced to surface-level text overlap, the ARHAHA shared task adopts a two-stage evaluation pipeline. The first stage performs automatic validation to enforce structural and lexical constraints. The second stage applies human evaluation to assess humor quality, originality, and safety. This separation ensures that only constraint-compliant outputs proceed to the more costly human assessment.

5.1 Automatic Validation

5.1.1 Automatic Evaluation Framework

To ensure consistent and scalable evaluation of system submissions, we implemented an automatic evaluation pipeline integrated with the Codabench platform. Participants were required to submit their generated outputs through the Codabench competition interface, where evaluation was performed automatically upon submission.

The evaluation pipeline is implemented as a Python scoring script that runs directly on the Codabench evaluation server. Once a submission file is uploaded, the platform automatically triggers the evaluation script, which processes the submitted predictions and computes the evaluation metrics. This setup ensures that all submissions are evaluated under identical conditions without manual intervention.

The automatic evaluation stage focuses on verifying task constraint compliance, including lexical requirements and formatting correctness, before proceeding to human evaluation.

5.1.2 Submission Format

Participants were required to submit their predictions as a ZIP archive containing a CSV file

with the generated outputs. Each row in the file corresponds to one dataset instance and includes the following fields:

- **id** – unique identifier of the prompt
- **generated_text** – the generated humorous text

The order of predictions must exactly match the order of the prompts in the provided dataset. Submissions that fail to follow the required format are rejected by the evaluation script.

5.1.3 Automatic Validation Criteria and Scoring

The scoring script evaluates each generated output according to several criteria designed to ensure compliance with the task constraints.

- **Word Inclusion Check:** The script verifies that the generated text contains the two required words associated with each prompt. The evaluation script performs a lexical matching check to confirm the presence of both required words in the generated output. Outputs are classified into three categories: Both words present, only one word present and neither word present. Only outputs containing both required words are considered fully compliant with the lexical constraint.
- **Length Constraint Verification:** The task imposes a strict maximum length constraint of 100 characters, including white spaces. The evaluation script automatically checks the length of each generated text. Outputs exceeding the maximum allowed length are flagged as length violations and penalized accordingly.
- **Format Compliance:** The evaluation script verifies the submission structure, ensuring that the number of generated texts matches the number of prompts, all required fields are present, and the submission file follows the expected format. Submissions failing these checks receive a score of zero until the formatting errors are corrected.

The automatic compliance score is computed as:

$$\text{Compliance Score} = \frac{1}{N} \sum_{i=1}^N v_i$$

where N is the total number of prompts, and v_i indicates whether the generated output for prompt i is valid ($v_i = 1$ if the output contains both required words and satisfies the length constraint, and 0 otherwise).

5.1.4 Integration with Codabench

The evaluation script is integrated directly with the Codabench competition framework, allowing automatic scoring of submissions. When a participant uploads a prediction file, Codabench automatically executes the evaluation script and returns the results to the leaderboard.

This evaluation infrastructure ensures transparent and reproducible scoring, immediate feedback for participants, and consistent evaluation across all submissions.

The automatic evaluation stage serves as a preliminary validation step, ensuring that generated outputs satisfy the structural constraints of the task before they are considered in the human evaluation phase.

5.2 Human Evaluation

Because humor quality cannot be reliably captured by automatic metrics alone, we complement the automatic validation with a human evaluation. Three independent native Arabic speakers assessed a subset of the generated jokes across four qualitative dimensions: humor, originality, lexical compliance and safety.

5.2.1 Evaluation Dimensions

The human evaluation considers four main dimensions, each targeting a distinct aspect of output quality:

1. **Humor Quality.** Annotators determine whether the generated text is perceived as humorous (binary: yes/no).
2. **Originality.** Annotators rate the novelty of each joke on a five-point scale, where 1 indicates a directly recycled joke and 5 indicates a fully novel humorous construction. This scale was designed to distinguish between recycled humor and genuinely creative output.
3. **Lexical Constraint Compliance.** Annotators verify whether the generated joke contains the required input words. Outputs are classified into three categories: both words present, only one word present, or neither word present.
4. **Safety and Policy Compliance.** Annotators flag any outputs containing policy violations (e.g., hate speech, offensive content, or personal attacks) or harmful stereotypes targeting individuals, groups, or cultures.

5.2.2 Annotation Setup and Procedure

The evaluation is conducted using a custom annotation interface implemented in Label Studio. A screenshot of the annotation interface is provided in Appendix A (Figure A1). Each participating system generated 1200 jokes, from which a subset of 300 jokes per system was selected for human assessment. To ensure a balanced and representative evaluation, we adopted a stratified sampling strategy rather than purely random sampling. The dataset is organized into ten semantic categories, each designed to capture a different type of humor-inducing scenario. From each category, 30 jokes were selected, resulting in a total of 300 jokes per system. This approach ensures that all humor categories are equally represented in the evaluation subset and prevents bias toward any particular type of humor. By maintaining a uniform

distribution across categories, the evaluation provides a more reliable and comprehensive assessment of system performance across diverse humor types. Each joke was independently evaluated by three annotators, with no access to participants' identities. All annotators were native Arabic speakers. Prior to the main annotation phase, annotators completed a calibration session in which they independently evaluated a small set of sample jokes and discussed disagreements to align their understanding of the evaluation criteria.

For each joke, annotators were presented with the generated text alongside the required word pair and assessed all four evaluation dimensions. Binary judgments (e.g., humor detection and policy violations) were aggregated using majority voting, while numerical ratings (e.g., originality) were averaged across annotators.

5.2.3 Scoring Formula

The final human score is computed as a weighted combination of the four evaluation dimensions. The weights were selected to reflect the primary objective of the task, which is generating humorous content. Humor was assigned the highest weight (0.4) as it represents the core task objective. Originality (0.3) was emphasized to encourage creative generation, while lexical compliance (0.2) ensures adherence to task constraints. Safety (0.1) was included as a necessary constraint rather than a primary objective, given the low violation rates observed across systems. These weights were determined based on task design priorities rather than learned or optimized parameters:

$$\text{Final Score} = 0.4 \times \text{Humor} + 0.3 \times \text{Originality} + 0.2 \times \text{Lexical Compliance} + 0.1 \times \text{Safety}$$

where Humor is the percentage of outputs judged humorous by majority vote, Originality is normalized to a percentage scale by dividing the mean score by 5 and multiplying by 100, and Lexical Compliance is measured by the proportion of outputs containing both required words. Safety is defined as the complement of violation and stereotype rates:

$$\text{Safety} = \max(0, 100 - (\text{Violation Rate} + \text{Stereotype Rate}))$$

where Violation Rate and Stereotype Rate are the percentages of outputs flagged for policy violations and harmful stereotypes, respectively. This formulation treats Safety as the complement of the total penalty incurred across both safety criteria. A system with no violations and no stereotyping receives a perfect Safety score of 100. If the combined violation and stereotype rates reach or exceed 100%, the Safety score is clamped to 0, reflecting that any further distinction among unsafe systems is not meaningful. The weights in the final score formula reflect the task's primary emphasis on generating genuinely funny content, while also rewarding creative novelty, adherence to the lexical constraints that

distinguish this task from open-ended generation, and compliance with ethical guidelines.

5.2.4 Inter-Annotator Agreement

Inter-annotator agreement (IAA) was assessed among three independent annotators across five evaluation dimensions: humor detection (binary), originality (ordinal, 1–5 scale; see below for scale calibration), stereotype presence (binary), cultural violation (binary), and keyword inclusion check (3 categories). A total of 900 jokes (300 per participating system) were evaluated, with each joke rated by all three annotators.

We report four complementary IAA metrics to ensure a thorough assessment of annotation consistency. Table 1 provides the interpretive scales used for the three chance-corrected metrics.

Range	Fleiss' κ	Krippendorff (α)	Gwet (AC1)
< 0.00	Poor	—	Poor
0.00–0.20	Slight	—	Slight
0.21–0.40	Fair	—	Fair
0.41–0.60	Moderate	—	Moderate
0.61–0.80	Substantial	Tentative (≥ 0.667)	Substantial
0.81–1.00	Almost Perfect	Reliable (≥ 0.800)	Almost Perfect

Table 1: Interpretive scales for chance-corrected IAA metrics.

Percentage agreement is computed as the proportion of items on which all annotators assign the same label (Cohen, 1960). It provides a transparent baseline but does not account for chance agreement and can be inflated when label distributions are highly skewed.

Fleiss' κ (Fleiss, 1971) corrects for chance agreement and extends Cohen's kappa to three or more annotators. It benefits from the Landis and Koch (1977) interpretive scale. However, it is sensitive to the kappa paradox (Feinstein & Cicchetti, 1990), where extreme label prevalence inflates expected chance agreement and yields paradoxically low κ values despite high actual consistency.

Krippendorff's α (Krippendorff, 2011) supports ordinal measurement scales through distance-weighted disagreement, which is important for the originality dimension, and handles missing data natively. Krippendorff recommends $\alpha \geq 0.800$ as reliable and $\alpha \geq 0.667$ as tentatively acceptable. Like Fleiss' κ , it is susceptible to the kappa paradox.

Gwet's AC1 (Gwet, 2008) addresses the kappa paradox by estimating chance agreement based on the probability of informed decisions rather than observed label distributions, making it robust under extreme prevalence. We use AC1 as our primary interpretive metric for skewed

dimensions, while retaining κ and α for cross-study comparability.

6. Baseline System

Unlike many shared tasks, ARHAHA does not provide an official baseline system. To our knowledge, no established model for constrained Arabic humor generation currently exists that could serve as a meaningful reference, as prior work on computational humor has focused predominantly on English and on detection rather than generation tasks.

The absence of a prescribed baseline allows participating teams to freely experiment with a wide range of generative modeling approaches, including fine-tuned language models, structured prompting strategies, and retrieval-augmented generation pipelines. Rather than constructing an ad hoc baseline that might anchor participants toward a particular modeling approach, we assess system performance directly through the evaluation framework described before.

We acknowledge the absence of a baseline as a limitation and intend to introduce a standardized baseline in future iterations, such as a zero-shot prompted large language model with no task-specific fine-tuning, to provide a minimum performance reference for comparison.

7. Participating Systems

The ARHAHA shared task attracted participation from several research teams interested in Arabic natural language generation and computational humor. In total, nine teams registered for the competition through the Codabench platform. Among the registered teams, three teams submitted final system outputs during the official evaluation phase *astral_fate* (Fatimah, 2026), *PassantElchafei* (Passant et al., 2026), and *jana-sami*. Out of these, two teams submitted system description papers providing detailed descriptions of their proposed methods and system architectures, while *jana-sami* submitted outputs but did not provide a system description paper. The attrition from registration to submission may reflect the difficulty of constrained humor generation and resource constraints.

The three systems explored different approaches for constrained Arabic humor generation. These approaches primarily relied on large language models and fine-tuned transformer architectures, reflecting recent trends in generative NLP research. Some systems adopted structured prompting strategies and multi-stage generation pipelines, while others focused on training compact generative models using curated humor datasets and synthetic data augmentation.

The submitted systems vary in their modeling strategies, including fine-tuning pretrained language models, parameter-efficient adaptation techniques, and retrieval-augmented generation pipelines.

Detailed descriptions of the participating systems can be found in the corresponding system description papers submitted by the teams. The following section introduces the official results of the ARHAHA shared task.

8. Results

Table 2 summarizes the final human evaluation results. The results show that *astral_fate* ranked first with a final human score of 49.25, *PassantElchafei* ranked second with a score of 44.62, and *jana-sami* ranked third with a score of 35.99.

The most striking finding is the very low humor rate across all three systems. Only 5.01% of *astral_fate*'s outputs, 4.34% of *PassantElchafei*'s outputs, and 0.33% of *jana-sami*'s outputs are judged humorous by the annotators. This indicates that generating genuinely funny content remains the dominant challenge across all participating systems in constrained Arabic humor generation, even when systems successfully satisfy lexical and safety constraints.

On originality, *astral_fate* and *PassantElchafei* score comparably (60.31% and 60.51%, respectively), while *jana-sami* scores substantially lower (20.00%), reflecting that its outputs were consistently rated as unoriginal by all annotators. All three systems demonstrate strong adherence to safety constraints, with very low violation rates (0.67% for *astral_fate* and *PassantElchafei*, 0.00% for *jana-sami*) and minimal stereotyping (0.44%, 0.11%, and 0.78%, respectively), yielding near-perfect Safety scores (98.89%, 99.22%, and 99.22%, respectively).

Lexical compliance remains a key differentiator across the three systems. *astral_fate* and *jana-sami* achieve near-perfect lexical compliance, correctly including both required words in 96.33% and 99.67% of evaluated outputs, respectively. By contrast, *PassantElchafei* includes both words in only 74.05% of outputs. Table 3 provides a detailed breakdown: the proportion of outputs missing both required words is negligible for *astral_fate* (0.11%) and *jana-sami* (0.33%) but substantial for *PassantElchafei* (21.94%), while the one-word partial compliance rates are 3.56% for *astral_fate*, 4.01% for *PassantElchafei*, and 0.00% for *jana-sami*.

The overall inter-annotator agreement reported in Table 4 was computed across all evaluated jokes from all participating systems as a single pooled dataset ($N = 900$ jokes, 3 annotators). Across all five dimensions, percentage agreement ranged from 94.35% to 99.25%, and Gwet's AC1 ranged from 0.94 to 0.99, indicating almost perfect agreement. This suggests that the low humor rates are not due to annotator subjectivity, but rather reflect a consistent and shared judgment that the generated outputs fail to be humorous.

Keyword inclusion check and originality are the dimensions where all four metrics converge on

high values. Keyword check achieves $\kappa = 0.80$, $\alpha = 0.80$, AC1 = 0.96, while originality achieves $\kappa = 0.93$, $\alpha = 0.97$, AC1 = 0.97.

Humor detection, stereotype, and violation all show high percentage agreement ($\geq 94\%$) and high AC1 (≥ 0.94), yet low or near-zero κ and α . This is due to the kappa paradox (Feinstein & Cicchetti, 1990), when one category overwhelmingly dominates (e.g., 96.8% not funny, 99.3% not stereotypical), kappa-family metrics estimate that nearly all agreement is attributable to chance. This skew reflects genuine properties of the generated content, as the majority of jokes were independently and consistently judged as not funny, not stereotypical, and not offensive by all three annotators. Gwet's AC1 confirms the agreement is genuine (0.94–0.99).

9. Analysis of Results

The results reveal a fundamental challenge in constrained Arabic humor generation: while systems can reliably satisfy lexical constraints and safety requirements, they largely fail to produce content that is perceived as genuinely humorous. With humor rates of 5.01%, 4.34%, and 0.33%, most generated outputs are fluent and structurally valid but not funny. This suggests that current generative models can follow surface-level instructions but lack the deeper capacity for incongruity, surprise, and culturally grounded comedic reasoning that humor requires.

A category-level analysis further confirms that this limitation is not confined to specific types of humor. Across all ten semantic categories, humor rates remain near zero, with most categories exhibiting less than 1.2% humorous outputs. This indicates that the difficulty of humor generation is not category-specific, but reflects a general limitation of current models across diverse semantic contexts. This finding is further supported by consistently high inter-annotator agreement across categories, suggesting that the absence of humor is reliably identified rather than a result of subjective disagreement.

Originality scores for *astral_fate* and *PassantElchafei* are closely comparable (60.31% and 60.51%), while *jana-sami* scores substantially lower (20.00%). *jana-sami*'s outputs were consistently rated as unoriginal, suggesting heavy reliance on recycled or templated content. Despite comparable originality, neither *astral_fate* nor *PassantElchafei* achieves high humor rates, confirming that novelty alone is not sufficient for humor. A joke can be original without being funny, pointing to the importance of narrative structure and punchline delivery.

Lexical compliance varies substantially across systems. *jana-sami* and *astral_fate* achieve near-perfect word inclusion (99.67% and 96.33%), while *PassantElchafei* omits both required words in over 21% of outputs. *jana-sami*'s high lexical compliance and safety scores are offset by its very low originality (20.00%) and humor (0.33%),

resulting in the lowest final score despite strong constraint adherence. This illustrates that constraint satisfaction alone does not produce competitive performance when the core task objective is humor generation.

All three systems demonstrate that safety constraints can effectively regulate generative behavior without eliminating creativity. Violation rates are very low (0.67% for *astral_fate* and *PassantElchafei*, 0.00% for *jana-sami*), and stereotyping remains minimal across all three systems (0.44%, 0.11%, and 0.78%). The near-perfect Safety scores (98.89%, 99.22%, and 99.22%) confirm that the ethical guidelines imposed by the task are achievable and do not suppress the generation of humorous content. This is a particularly important finding for computational humor research, where safety and appropriateness remain central concerns.

Overall, the results indicate that the dominant bottleneck in constrained humor generation is humor itself, not safety, originality, or constraint compliance. The strongest system is the one that performs well across all four evaluation dimensions, but all systems remain far from consistently producing genuinely humorous content.

Moreover, to better understand the nature of humor generation failures, we examine representative examples of low-quality outputs; you can see the qualitative error analysis in Appendix B.

10. Discussion

The findings highlight that constrained Arabic humor generation remains a substantially unsolved problem. The very low humor rates across all three systems, below 6%, indicate that current generative models are not yet capable of reliably producing content that humans find funny, even when they successfully satisfy lexical and safety constraints. This stands in contrast to the other evaluation dimensions, where all three systems perform well, and suggests that humor represents a qualitatively different challenge from constraint adherence or content safety.

This limitation is further corroborated by the category-level analysis, which indicates that humor rates remain uniformly low across all semantic categories. This suggests that the difficulty is not confined to specific types of humor but reflects a general limitation of current models across diverse contexts.

This observation has practical implications for system design. Current approaches appear capable of generating content that is safe,

original, and lexically grounded, but fundamentally struggle with the core task of being funny. Future systems may need to move beyond standard language modeling objectives and incorporate explicit humor-aware training signals, such as reward models trained on human humor judgments, punchline structure modeling, or incongruity detection mechanisms. Additionally, constrained decoding strategies and post-generation filtering may help improve lexical compliance further but are unlikely to address the humor deficit without deeper architectural changes.

Beyond system performance, the divergence between kappa-family metrics and Gwet's AC1 reveals an important consideration for evaluating humor generation tasks. Because most generated jokes are genuinely not funny, any binary humor annotation will inevitably produce a heavily skewed distribution, making traditional agreement metrics unreliable for this type of evaluation. This suggests that future humor generation shared tasks should adopt prevalence-robust metrics such as Gwet's AC1 as the primary measure of annotation quality, rather than relying solely on Fleiss' κ or Krippendorff's α . Originality and keyword check avoid this issue: originality benefits from clear separation between systems rated as unoriginal (score 1) and those rated as somewhat original (score 3), while keyword check benefits from a three-category distribution with sufficient spread for kappa-family metrics.

More broadly, the results suggest that future work in Arabic humor generation should focus on modeling incongruity, punchline structure, and culturally appropriate humor mechanisms rather than fluency or constraint adherence alone. The challenge is not generating grammatically correct Arabic text that contains two required words. It is making that text genuinely funny. This underscores that humor generation is not only a problem of language modeling, but also of capturing subtle cognitive and cultural mechanisms that remain poorly understood even in computational terms.

The task also demonstrates the value of combining automatic validation with human evaluation. Automatic checks enforce structural requirements efficiently but cannot capture humor quality or originality. Human evaluation provides a richer picture of system behavior. Together, these two components form a robust and reproducible evaluation framework for computational humor generation. Future iterations of the task may benefit from introducing a standardized baseline, expanding the annotator pool, and exploring more fine-grained agreement metrics to further strengthen the evaluation methodology

Metric	astral_fate (%)	PassantElchafei (%)	jana-sami (%)
Humor	5.01	4.34	0.33
Originality	60.31	60.51	20.00
Lexical Compliance	96.33	74.05	99.67
Safety	98.89	99.22	99.22
Violations	0.67	0.67	0.00
Stereotypes	0.44	0.11	0.78
Final Score	49.25	44.62	35.99

Table 2: Final human evaluation results for ARHAHA 2026.

	astral_fate (%)	PassantElchafei (%)	jana-sami (%)
Both Words	96.33	74.05	99.67
One Word	3.56	4.01	0.00
Missing Words	0.11	21.94	0.33

Table 3: Lexical compliance breakdown. Both Words corresponds to the Lexical Compliance score in Table 2. One Word and Missing Words show partial and complete constraint failures, respectively.

Dimension	Scale	% Agreement	Fleiss' κ	Kripp. α	Gwet AC1	Interpretation*
Humor Detection	Binary	94.35	0.10	0.10	0.94	Almost Perfect.
Originality	Ordinal (1–5)	96.92	0.93	0.97	0.97	Almost Perfect.
Stereotype	Binary	99.10	−0.01	−0.01	0.99	Almost Perfect.
Violation	Binary	99.25	0.16	0.16	0.99	Almost Perfect.
Keyword Check	3 categories	96.40	0.80	0.80	0.96	Almost Perfect.

Table 4: Overall inter-annotator agreement per evaluation dimension. * Interpretation based on Gwet's AC1, which is robust to the kappa paradox.

11. Conclusion

This paper presented ARHAHA 2026, the first shared task on constrained Arabic humor generation. The task requires systems to generate humorous Arabic text incorporating a given pair of words while adhering to safety and cultural guidelines. Through a two-stage evaluation pipeline combining automatic validation with human assessment, we evaluated three participating systems across four dimensions: humor quality, originality, lexical compliance, and safety.

The main findings are threefold. First, humor generation is the dominant bottleneck. All three systems achieve very low humor rates, indicating that producing genuinely funny content remains far beyond the reach of current approaches, even when other constraints are reliably met. Second, lexical compliance and safety are comparatively achievable, with the top system reaching near-perfect scores on both. Third, the strongest overall performance comes from performing well across all four evaluation dimensions, but all systems remain far from the goal of consistent humor generation.

The task has several limitations that should be acknowledged. No official baseline is provided, making it difficult to contextualize absolute performance levels. The number of participating systems is small, limiting the generalizability of the findings.

ARHAHA provides a valuable benchmark for advancing research in Arabic humor generation and controlled text generation more broadly. Future iterations of the task will explore more sophisticated evaluation metrics, improved modeling of humor structures, and better integration of cultural and contextual knowledge in generative systems. More broadly, the findings suggest that humor generation is not merely a problem of fluent language production, but a deeper challenge involving semantic reasoning, incongruity, and cultural understanding that remains largely unresolved in current NLP systems.

12. Ethics Statement

This work involves human evaluation of generated text. Annotators participated voluntarily and were informed of the task objectives. No personal or sensitive data were used. Annotators were instructed to flag any harmful or offensive content.

13. Limitations

This study has several limitations. The number of participating systems is limited, and humor evaluation remains inherently subjective. The task focuses on short-form humor under lexical

constraints, which may not fully reflect natural humor.

14. References

- Raskin, V. (1985). *Semantic Mechanisms of Humor*. Dordrecht: Springer.
- Mihalcea, R., and Strapparava, C. (2005). Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, 531–538.
- Castro, S., Chiruzzo, L., Góngora, S., Rahili, S., Deng, N., Sastre, I., Amoroso, V., Rey, G., Rosá, A., Moncecchi, G., Meaney, J. A., Prada, J. J. and Mihalcea, R. (2026). SemEval-2026 Task 1: MWAHAHA, Models Write Automatic Humor And Humans Annotate. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46.
- Feinstein, A. R. and Cicchetti, D. V. (1990). High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6), 543–549.
- Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5), 378–382.
- Gwet, K. L. (2008). Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1), 29–48.
- Krippendorff, K. (2011). Computing Krippendorff's alpha-reliability. Annenberg School for Communication Departmental Papers.
- Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 159–174.
- Fatimah Eldin. 2026. Do3aba-AI at AraHAHA Shared Task 2026: Bridging the Gap in Arabic Humor Generation via Synthetic Data Distillation. In *Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks (OSACT 2026) co-located with the 2026 International Conference on Language Resources and Evaluation (LREC2026)*, Palma, Mallorca (Spain).
- Passant Elchafei, Amany Fashwan, Maha Ezzat and Mohamed Abdelbaset. 2026. Arabic-HuCoT at ARHAHA 2026 Shared Task: Arabic Humor Generation with ALLaM and HuCoT-Based Structured Prompting. In *Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks (OSACT 2026) co-located with the 2026 International Conference on Language Resources and Evaluation (LREC2026)*, Palma, Mallorca (Spain).

Appendix A

The Joke | النكتة 🗨️
جاءت دراجة للمقهى وقالت: "أنا مش عابرة كحل، عابرة كافي!"
الكلمات المطلوبة | Required Words: دراجة ومقهى

?Is it funny | هل هي مضحكة؟ 😄
Yes, funny 😄 | نعم، مضحكة
No, not funny 😞 | لا، ليست مضحكة

Originality Score | درجة الأصالة 💡
هل النكتة جديدة أم مكررة؟ | Is the joke original or recycled?
1 - مكررة جداً | Very recycled
2 - مألوفة | Familiar
3 - جديدة نوعاً ما | Somewhat original
4 - أصيلة | Original
5 - مبتكرة جداً | Highly innovative

✓ تحتوي الكلمتين المطلوبتين | Contains required words
الكلمات موجودتان | Both words present
كلمة واحدة فقط | Only one word
لا توجد أي منهما | Neither word present

⚠️ أي انتهاك للقواعد | Any policy violation
لا يوجد انتهاك | No violation
يوجد انتهاك | Violation

هل تحتوي قوالب نمطية؟ | Contains stereotyping
No | لا
Yes | نعم

Figure A1: Annotation interface used for human evaluation, implemented in Label Studio

Appendix B

Qualitative Error Analysis

To better understand the nature of humor generation failures, we examine representative examples of low-quality outputs. Many generated jokes are syntactically correct but lack semantic coherence or a clear humorous structure.

For example: "واحد غبي شاف الشمس بتتحرك ببطء قال يا ريت "البطيخة تتحرك زيها عشان ما أروحهاش بدري كل يوم".

This example breaks down because it fails to establish a meaningful semantic connection between the watermelon and going to work. As a result, the attempted incongruity remains ungrounded and does not develop into a coherent humorous effect.

Similarly: "شاحن الهاتف عنده مشكلة في طهي الطعام، لانه بيشتحن "الأكل زي التليفون".

While this example attempts a form of analogy, it fails to produce a clear incongruity or punchline, leading to a weak and unconvincing humorous construction.

These examples illustrate that current systems often generate text that is grammatically well-formed but semantically incoherent or lacking the structural elements required for humor, such as setup, incongruity, and resolution.

Another recurring failure pattern involves incorrect interpretation of ambiguous words. In several

cases, models misinterpret the intended sense of a word, leading to incoherent or unintended outputs.

For example: "استاذ الفلسفة قال لصوص في الشارع: 'أنت بتحب "إلصوصة، بس مش بتفهمها'"

In this case, the required word "استاذ" is intended to refer to a stadium, which aligns with the humor category (e.g., size contrast or physical context). However, the model interprets it as "استاذ" (teacher/professor), resulting in a semantically incorrect setup. This misinterpretation breaks the intended context and prevents the joke from forming a coherent or meaningful humorous structure.

Similarly: "مرة واحد فتح الطاقة الشمسية في بيت مظلم قال له "صاحبه لماذا؟ قال عشان أظلمها من جديد".

The required word "ظلم" is intended to mean injustice, but the model shifts toward a different semantic field related to darkness (ظلام), resulting in a confused or incorrect usage. This indicates a failure to distinguish between semantically related but distinct concepts, leading to a breakdown in meaning and humor.

These errors highlight a limitation in handling lexical ambiguity, where models fail to correctly disambiguate word meanings based on context, which is critical for constructing coherent and effective humor.

Another observed pattern involves implicit substitution of required words. Instead of explicitly including the required lexical items, the model generates text that conveys their meaning implicitly, thereby violating the task constraints despite partial semantic understanding.

For example: "مدير الشركة قابل موظف جديد وقال له: 'أنا مدير' "إلهنا، الموظف رد عليه: 'أنا مدير بيتي'"

The required phrase "مناقشة حادة" is not explicitly included. Instead, the model implies a form of confrontation through dialogue, without using the required lexical expression. This suggests that the model captures the semantic intent but fails to satisfy the explicit constraint.

For example: "ابن الجيران: إيه ده؟ ليه كل مره أروح عندكم أجد "بروست في الفريزر"

The required phrase "استفسار غريب" is represented through a question, but not explicitly included, indicating reliance on implicit expression instead of direct lexical realization.

These examples reveal that models may partially understand the intended semantic concepts, but fail to adhere to strict lexical constraints, highlighting a gap between semantic generation and constraint-controlled generation.

The AdabEval 2026 Shared Task on Arabic Politeness Detection

Reem Alqifari¹, Hend Al-Khalifa¹, Nadia Ghezaiel Hammouda², Maria Bounnit³
Hend AlHazmi⁴, Ameer Almasoud¹, Sharefah Al-Ghamdi¹, Noof Alfear¹

¹College of Computer and Information Sciences, King Saud University,

²College of Computer Science and Engineering, University of Hail,

³Cadi Ayyad University, ⁴Saudi Center of Philosophy and Ethics

¹Riyadh, Saudi Arabia, ²Hail, Saudi Arabia,

³Marrakesh, Morocco, ⁴Jeddah, Saudi Arabia

¹ralgify|hendk|ammalmasoud|sharefah|nalfeared@ksu.edu.sa,

²ghezaielnadia.ing|³mariabounnit|⁴hend.hamed.w@gmail.com

Abstract

We present an overview of the AdabEval 2026 shared task, organized as part of the OSACT7 workshop (co-located with LREC 2026). This task introduces the first benchmark suite for politeness detection. It includes two subtasks: Politeness Classification (Subtask A) and Category Prediction (Subtask B). The task focuses on evaluating models' ability to recognize and categorize politeness phenomena in Arabic text. Evaluation was conducted using an automatic metric (macro F1-score). A total of 28 unique teams participated in the shared task. Of these, 13 teams submitted final system predictions across the two subtasks. The top-performing systems relied primarily on transformer-based architectures. The winning systems achieved macro F1-scores of 0.89 for Subtask A and 0.58 for Subtask B.

Keywords: Politeness Classification, Politeness Category Prediction, Arabic Natural Language Processing

1. Introduction

Politeness plays a central role in conversation, languages provide diverse ways to encode respect and reduce imposition, and these markers are intimately tied to social power and interpersonal harmony. In Arabic, the concept is deeply rooted in cultural norms; speakers employ honorifics, plural forms of address and kinship terms to show respect. Most computational work on politeness focuses on English, with only limited studies on other languages. The ADAB dataset introduced the first large-scale Arabic politeness dataset, containing 10,000 annotated examples across three politeness categories (Al-Khalifa et al., 2026). The dataset used in this shared task is derived as a subset of ADAB. To the best of our knowledge, this shared task represents the first Arabic benchmark focused on politeness analysis, addressing both politeness prediction and the identification of politeness categories.

The shared task consists of two complementary subtasks: **Subtask A: Politeness Classification**, which focuses on identifying the overall politeness level of a text, and **Subtask B: Politeness Category Prediction**, which aims to identify the underlying pragmatic strategies expressed in the text. This distinction reflects the fact that politeness involves not only surface-level classification but also multiple interacting communicative functions within the same utterance.

Modeling politeness in Arabic presents several

challenges. Politeness strategies are often expressed implicitly and may rely on cultural conventions, formulaic expressions, or contextual cues. In addition, multiple categories may co-occur within a single sentence, making fine-grained prediction particularly challenging.

This overview paper describes the design and organization of the AdabEval shared tasks. We present the task definitions, dataset, evaluation protocol, baseline systems, and a summary of participants results. We conclude with insights gained from the shared task and directions for future research.

2. Task Description

The shared task consists of two subtasks. Subtask A focuses on building and evaluating models that automatically assess the politeness level of Arabic text. Given a text, participants must classify it into one of three categories: Polite, Neutral, or Impolite. Systems will be compared using macro-averaged F1-score on the test data. Subtask B evaluates the ability of systems to identify multiple pragmatic functions in Arabic social-media posts. Each text may express one or more categories from nine culturally grounded functions including: "Criticism", "Insult", "Respect", "Prayers", "Greetings", "Hospitality", "Gratitude", "Admiration_Love", and "Racism_Discrimination". The task is framed as multi-label classification, and systems must predict all applicable categories for each instance. The of-

Task	Train	Val	Test
Subtask A	4896	694	1407
Subtask B	2050	292	589

Table 1: Dataset split sizes for the AdabEval shared tasks.

ficial evaluation metric is macro-averaged F1-score across the nine categories.

3. Dataset Description

The dataset used in this shared task is derived as a subset of the ADAB dataset (Al-Khalifa et al., 2026), a large-scale Arabic politeness corpus. The subset was constructed to support the two subtasks, focusing on politeness classification and fine-grained politeness category prediction. Both subtasks rely on sentences collected from multiple online platforms and annotated according to a unified annotation scheme. Each sentence in the dataset is associated with a source platform indicating where the text was originally collected. The sources include YouTube comments, Shein product reviews, Twitter posts, and company product reviews. This diversity of sources allows the dataset to capture different styles of user-generated Arabic text across multiple communication contexts, including the natural mixture of Modern Standard Arabic (MSA) and dialectal Arabic commonly found in online communication.

3.1. Subtask A: Politeness Classification

Subtask A focuses on classifying sentences into three politeness classes: *Polite*, *Impolite*, and *Neutral*. Each instance in this task contains three fields: the sentence text, the source platform, and the corresponding politeness label.

The dataset is divided into training, validation, and test splits. The training set contains 4,896 instances, the validation set contains 694 instances, and the test set contains 1,407 instances. To ensure fair evaluation, the test set provided to participants includes only the sentence text and source information, and participants are required to predict the politeness label.

Table 1 summarizes the dataset splits for Subtask A, while the label distribution is illustrated in Figure 1.

3.2. Subtask B: Category Prediction

Subtask B focuses on identifying the politeness category expressed in sentences labeled as either *Polite* or *Impolite*. For this task, a subset of the original dataset was selected that contains only sentences belonging to these two classes.

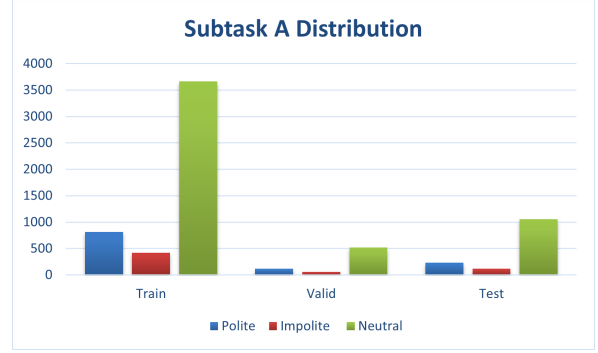


Figure 1: Distribution of politeness labels in the Subtask A

Each instance in the training and validation sets includes the following fields: the sentence text, the source platform, the politeness label, and up to four annotated politeness criteria. Each criterion is accompanied by a keyword that represents the lexical trigger associated with the identified strategy. These annotations were originally introduced to capture pragmatic signals underlying politeness expressions. For consistency in formatting, the dataset is represented using four columns (criteria1–criteria4), where unused fields are left empty if fewer than four criteria apply.

The dataset is split into training, validation, and test sets. The training set contains 2,050 instances, the validation set contains 292 instances, and the test set contains 589 instances. Similar to Subtask A, the test categories are hidden from participants and only the sentence text, source and label information are provided. Table 1 summarizes the dataset splits for Subtask B.

In addition to the binary politeness labels, each sentence in Subtask B is associated with a fine-grained politeness strategy category. These categories capture different pragmatic functions such as respect, criticism, gratitude, and prayers. Table 2 and Figure 2 present the distribution of politeness categories across the training, validation, and test splits. The dataset contains 2,904 instances distributed across several polite and impolite strategies. Overall, polite strategies are more frequent than impolite ones. The most frequent category is Respect (1,034 instances), followed by Criticism (637 instances). The category Prayers appears in both the polite and impolite classes. Additionally, some categories such as Racism_Discrimination and Hospitality, contain no instances in the current dataset due to the natural distribution of the collected data.

Category	Politeness	Train	Valid	Test
Criticism	Impolite	435	78	124
Insult	Impolite	155	22	64
Racism_Discrimination	Impolite	0	0	0
Prayers	Impolite	58	2	12
Prayers	Polite	357	57	95
Respect	Polite	741	105	188
Greetings	Polite	79	16	17
Hospitality	Polite	0	0	0
Gratitude	Polite	168	16	43
Admiration_Love	Polite	43	8	21

Table 2: Distribution of politeness categories across dataset splits.

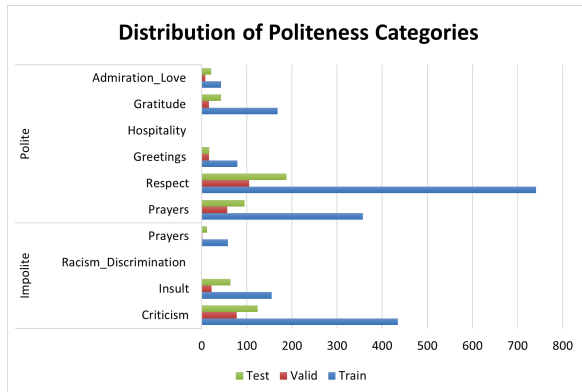


Figure 2: Distribution of politeness categories across dataset splits in Subtask B.

4. Evaluation

This section describes the evaluation protocol and submission procedures used in the AdabEval shared tasks. The evaluation was conducted using the Codabench platform¹, which allows participants to submit their system predictions and automatically evaluates them against the hidden test labels.

4.1. Evaluation Metric

For both subtasks, system performance is evaluated using the macro-averaged F1 score. This metric computes the F1 score independently for each class and then averages the scores across all classes. Macro-F1 is particularly suitable for this task because it assigns equal importance to all classes regardless of their frequency in the dataset, thereby mitigating the effects of class imbalance.

For Subtask A, the evaluation considers three classes: *Polite*, *Neutral*, and *Impolite*. For Subtask B, the evaluation measures the ability of systems to correctly predict the politeness category associated with each sentence.

¹<https://www.codabench.org/>

4.2. Submission Format

Participants were required to submit their predictions through the Codabench platform as a single compressed ZIP file. The ZIP archive must contain the prediction files corresponding to the subtasks.

Subtask A. For Subtask A, the ZIP file must include a file named `subtask_A.csv`. The CSV file must contain exactly one column named `label`, where each row corresponds to a predicted politeness label for a test instance. The predicted label must be one of the following classes: *Polite*, *Neutral*, or *Impolite*. Each test instance must receive exactly one prediction, and the row order in the submission file must match the order of the instances in the provided test set.

Subtask B. For Subtask B, the ZIP file must include a file named `subtask_B.csv`. The CSV file must contain exactly four columns named `criterial`, `criteria2`, `criteria3`, and `criteria4`. Each column contains a predicted politeness strategy category, or is left empty if fewer than four categories apply. Each sentence may be assigned one or more categories, with a maximum of four categories per instance. Although each instance may have fewer than four labels, a fixed four-column format is used to standardize submissions.

Each row in the file corresponds to a test instance, and the row order must exactly match the order of the provided test data. Category labels are not mutually exclusive and must be selected from the official list of Subtask B categories. Submissions that do not follow the specified format may be rejected or receive a score of zero.

4.3. Evaluation Procedure

All submissions were evaluated using the Codabench platform. We implemented a custom scoring script that automatically computes the evaluation metric for each submitted prediction file. The scoring script reads the submitted files, validates their format, and compares the predictions against the hidden ground-truth labels of the test set.

For Subtask A, the script computes the macro-averaged F1 score over the three politeness classes. For Subtask B, the script evaluates the predicted politeness categories for each sentence and calculates the macro-averaged F1 score across all categories.

The scoring script also ensures that the number and order of predictions match the provided test data. Submissions that do not follow the required format are rejected or receive a score of zero.

Task	Val Macro-F1	Test Macro-F1
Subtask A	0.83	0.85
Subtask B	0.41	0.38

Table 3: Performance of the MARBERT baseline models.

4.4. Competition Phases

The shared task was organized into two phases to facilitate system development and fair evaluation. During the evaluation phase, participants were provided with the training and validation datasets to train and tune their models. Participants could submit predictions to the Codabench leaderboard and receive evaluation scores based on the validation set. In the final test phase, participants were required to submit their predictions for the hidden test set. The ground-truth labels for the test data were not released, and the final ranking of participating systems was determined based on their performance on the test set.

5. Baseline Systems

To provide a reference point for participating teams, we implemented baseline systems for both subtasks using a transformer-based Arabic language model. Specifically, we fine-tuned MARBERT (Abdul-Mageed et al., 2021), a pre-trained language model designed for Arabic social media text, on the training data of each subtask. MARBERT was selected as the baseline model due to its strong performance reported in prior work comparing a wide range of approaches, including traditional machine learning models, transformer-based models, and large language models (Al-Khalifa et al., 2026).

For Subtask A, the model was fine-tuned to perform multi-class classification over the three politeness labels. The model takes the input sentence and predicts one of the three classes.

For Subtask B, MARBERT was adapted to predict the politeness categories associated with each sentence. Since multiple categories may apply to a single sentence, the model was trained in a multi-label classification setting, allowing it to assign one or more strategy labels per instance.

The baseline models were trained using the official training splits provided for each task, and performance was evaluated on both the validation and test sets using the same evaluation metric described in Section 4.1. Table 3 reports the performance of the baseline systems for both subtasks.

6. Participating Systems

The AdabEval shared tasks attracted participation from multiple research teams working on Arabic

Rank	Team	Macro-F1
1	grkurdi*	0.89
2	Nina*	0.87
3	abdllh	0.87
4	rand_at	0.85
5	Baseline	0.85

Table 4: Leaderboard results for Subtask A. Teams marked with * submitted system description papers.

NLP. Different teams participated in each subtask, reflecting varying research interests in politeness classification and politeness category prediction.

A total of 28 unique teams registered for the AdabEval shared tasks across both subtasks.

For Subtask A, 19 teams registered, of which 8 submitted final system predictions. For Subtask B, 13 teams registered, of which 5 submitted final results.

These numbers are not mutually exclusive, as some teams participated in both subtasks. Participants were allowed to compete in one or both subtasks independently.

The majority of participating systems relied on transformer-based architectures, particularly Arabic pre-trained language models such as AraBERT (Antoun et al., 2020) and MARBERT. Some teams also explored multilingual transformer models and ensemble methods.

Detailed descriptions of the individual systems are provided in the corresponding system papers submitted by the participating teams. Several teams submitted system predictions but did not submit system description papers. Five teams submitted system description papers. In the next section of this overview paper, we report the official results of the shared tasks based on the evaluation of system predictions on the hidden test sets.

7. Results

This section presents the official leaderboard results for the AdabEval shared tasks. System performance was evaluated on the hidden test sets using the macro-averaged F1 score. Tables 4 and 5 report the results for Subtask A and Subtask B, respectively. Teams marked with an asterisk (*) submitted system description papers. The leaderboard tables report only systems that outperform the baseline model.

8. Discussion

The results provide several insights into the challenges of automatic politeness detection in Arabic. Overall, systems achieved stronger performance on Subtask A compared to Subtask B, reflecting

Rank	Team	Macro-F1
1	gfattani*	0.58
2	alla_zwawi*	0.55
3	jana-sami	0.55
4	rand_at*	0.41
5	Baseline	0.38

Table 5: Leaderboard results for Subtask B. Teams marked with * submitted system description papers.

the increased difficulty of predicting fine-grained politeness strategies.

For Subtask A, the results reveal a strong bias toward the neutral class. As the dominant category in the dataset, neutral instances were frequently predicted by the systems, leading to many polite and impolite utterances being misclassified as neutral. This bias suggests that models tend to favor the majority class when explicit politeness cues are absent, highlighting the difficulty of distinguishing subtle politeness signals from neutral expressions.

In Subtask B, performance varied across different categories. This variability can be attributed to several interrelated factors. Categories such as Respect and Prayers proved relatively straightforward to detect. This is largely because such expressions tend to follow recognizable formulaic patterns deeply embedded in Arabic communicative norms. Speakers of Arabic frequently employ fixed religious or honorific phrases in these contexts, providing models with reliable lexical anchors. In contrast, categories such as Criticism and Insult were more challenging because they often rely on contextual or implicit cues.

Furthermore, expressions categorized as Prayers can convey either positive or negative intent depending on context. Prayer expressions may function as polite wishes directed toward someone’s well-being (e.g., wishing someone success or good fortune) or as negative imprecations directed toward someone (e.g., wishing harm). Such contextual variation increases the ambiguity of the category and further complicates fine-grained politeness strategy prediction. The multi-label nature of Subtask B added yet another layer of complexity. Unlike single-label classification, where a model selects the most fitting category, multi-label settings require the simultaneous identification of potentially overlapping labels, demanding a more nuanced understanding of the input. Further contributing to lower performance was the issue of class imbalance. When certain categories appear only rarely in the training data, models struggle to learn meaningful representations for them, and this weakness persists even when compensatory techniques such as focal loss or class-weighted training are employed. This was evident in the notably lower scores observed for

less frequent categories, where even well-tuned systems failed to achieve reliable detection.

Across both subtasks, the participating systems converged on a broadly similar approach, relying on transformer-based architectures pretrained on Arabic text, particularly AraBERT and MARBERT. These models were fine-tuned using standard classification objectives, with some teams introducing modifications to handle class imbalance more effectively. The strong results observed in Subtask A affirm that pretrained models are well suited to capturing the more explicit and conventionalized markers of politeness in Arabic. However, the comparatively lower performance in Subtask B suggests that these models, despite their considerable capacity, still fall short when confronted with pragmatically subtle or culturally specific expressions.

These observations highlight the importance of contextual and pragmatic understanding when modeling politeness in Arabic. Politeness strategies are often conveyed through culturally specific expressions and indirect formulations that may not be captured by surface lexical cues alone.

9. Conclusion

This paper presented an overview of the AdabEval shared tasks, which aim to advance research on politeness understanding in Arabic. The shared task introduced two complementary subtasks: Subtask A focused on politeness classification into *Polite*, *Neutral*, and *Impolite*, while Subtask B addressed the prediction of fine-grained politeness categories. Both tasks were derived from a curated subset of a previously published Arabic politeness dataset, covering multiple sources of user-generated content.

The shared tasks attracted participation from multiple research teams, who explored a variety of modeling approaches, primarily based on transformer architectures and Arabic pre-trained language models. The leaderboard results demonstrate the progress made in modeling politeness in Arabic, while also highlighting the challenges associated with detecting subtle pragmatic cues and fine-grained politeness categories.

We hope that the AdabEval shared tasks will encourage further research on pragmatic and sociolinguistic aspects of Arabic NLP.

10. Limitations

Despite its contributions, the dataset and tasks have several limitations. First, the data are collected from a limited set of online platforms, which may not fully capture the diversity of Arabic language use across different domains, regions, and dialects.

Second, the dataset exhibits class imbalance. In Subtask A, the *Neutral* class constitutes the majority of instances, which may bias models toward predicting neutral labels. In Subtask B, certain categories (e.g., *Respect* and *Criticism*) are more frequent than others, leading to uneven category representation.

Third, some categories in Subtask B are underrepresented or absent in the training data. For example, categories such as *Hospitality* and *Racism_Discrimination* contain no instances in certain splits, which prevents models from learning reliable patterns for these categories.

Finally, the task is limited to sentence-level classification and does not incorporate broader discourse context, which may be necessary for accurately interpreting politeness in real-world communication.

Future work will aim to address these limitations by expanding the dataset to include more diverse sources, and exploring context-aware approaches to politeness modeling.

11. Ethics Statement

This shared task complies with the LREC 2026 Ethical Guidelines for responsible language resource development. It supports research on politeness understanding in Arabic while adhering to responsible data usage practices. The dataset is derived from publicly available user-generated content and has been anonymized to protect user privacy. No personally identifiable information was intentionally included.

All evaluations were conducted on hidden test labels, which were not accessible to participants, ensuring a fair and unbiased comparison across systems.

12. References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. [ARBERT & MARBERT: Deep bidirectional transformers for Arabic](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Hend Al-Khalifa, Nadia Ghezaiel, Maria Bounnit, Hend Hamed Alhazmi, Noof Abdullah Alfear, Reem Fahad Alqifari, Ameera Masoud Alma-soud, and Sharefah Ahmed Al-Ghamdi. 2026. Adab: Arabic dataset for automated politeness

benchmarking—a large-scale resource for computational sociopragmatics. *Proceedings of The 2026 International Conference on Language Resources and Evaluation (LREC2026)*. Palma, Mallorca, Spain.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15. European Language Resources Association. Marseille, France.

GHAD NLP at AdabEval2026: Transformer-Based Approach for Arabic Politeness and Pragmatic Category Classification

Ghader Kurdi and Ghada Alfattni

Department of Data Science, Department of Computer Science
{College of Computing, Jamoum University College}, Umm Al-Qura University, Makkah, Saudi Arabia
{grkurdi, gafattni}@uqu.edu.sa

Abstract

This paper presents our submission to the AdabEval 2026 shared task on Arabic politeness classification and pragmatic category prediction. We explored a range of Arabic-specific and multilingual transformer models and integrated their outputs through an ensemble strategy. Our approach achieved state-of-the-art performance in the shared task, ranking first in both subtasks with a macro-F1 score of 0.89 and an accuracy of 0.93 on subtask A, and a macro-F1 score of 0.58 on subtask B. Although our approach delivered high performance on overall politeness classification, pragmatic category prediction remains more challenging. Despite achieving the top ranking in this subtask, the comparatively lower macro-F1 score suggests that modelling fine-grained pragmatic functions requires further methodological refinement and experimentation.

Keywords: Politeness, Classification, Transformers

1. Introduction

Politeness is a fundamental aspect of human communication, shaping how speakers express respect, mitigate imposition, and maintain social harmony. In Arabic, politeness is deeply embedded in cultural and linguistic norms, where speakers employ honorifics, plural forms of address, kinship terms, prayers, and respectful expressions to convey social relationships and interpersonal attitudes. Automatic detection of politeness in Arabic social media is important for various applications such as content moderation and generation, sentiment analysis, dialogue systems, and conversational AI. However, most of the computational work on politeness has focused on English, with limited resources and research available for Arabic (Priya et al., 2024).

The AdabEval 2026 shared task (Alqifari et al., 2026) addresses this gap by introducing a large-scale annotated dataset of Arabic social media posts labelled for politeness and pragmatic functions, enabling systematic evaluation and comparison of computational approaches. The shared task comprises two subtasks: (A) politeness classification, which involves classifying text into polite, neutral, or impolite categories, and (B) category prediction, which requires identifying one or more pragmatic functions, such as criticism, insult, respect, prayer, or greeting. These tasks present unique challenges due to the diversity of Arabic dialects, informal writing styles, the brevity and limited contextual information characteristic of social media texts, and culturally grounded expressions of politeness. In this paper, we describe the GHAD (غَد) NLP team's submission to the AdabEval 2026 shared task. Our approach leverages pretrained Arabic

and multilingual transformer models fine-tuned for politeness classification and combines them using an ensemble strategy. Pretrained language models capture rich contextual and semantic information, making them particularly effective for pragmatic and sociolinguistic tasks. We explore multiple system configurations and investigate the impact of preprocessing strategies, training objectives, and model architectures. Our approach ranked first in both subtasks, attaining a macro-F1 score of 0.89 and an accuracy of 0.93 on subtask A, and a macro-F1 score of 0.58 on subtask B, highlighting its effectiveness.

Our contributions can be summarised as follows:

- we propose a transformer-based ensemble for Arabic politeness classification;
- we design a MARBERT-based multi-label framework with label-specific threshold optimisation;
- we conduct extensive experiments demonstrating state-of-the-art performance on both subtasks.

2. Background

2.1. Task Definition

The AdabEval 2026 shared task focuses on automatic detection of politeness and pragmatic functions in Arabic social media posts. The challenge consists of two subtasks, each addressing a different dimension of the classification challenge.

Subtask A: Politeness Classification. This subtask is formulated as a multi-class classification

problem. Given an Arabic text, the system must classify it into one of three politeness levels: *Polite*, *Neutral*, or *Impolite*. The goal is to capture the overall politeness expressed in the text.

Subtask B: Category Prediction. This subtask is formulated as a multi-label classification problem. Each text may express one or more pragmatic functions reflecting culturally grounded politeness strategies. The goal is to predict all applicable categories from a predefined set of nine categories: *Criticism*, *Insult*, *Disparagement*, *Prayers*, *Greetings*, *Admiration*, *Respect*, *Felicitation*, and *Hospitality & Generosity*. Examples of polite expressions include prayers and asking for permission, while impolite expressions include insults or criticism.

2.2. Dataset Description

The dataset, released as part of the AdabEval 2026 shared task and made available on Codabench,¹ consists of Arabic social media posts collected from four sources: Companies platforms, Shein, YouTube, and X. More details about the data sources as well as the collection and annotation processes can be found in (Al-Khalifa et al., 2026a). The dataset contains a total of 7,928 instances distributed across multiple files corresponding to different subtasks and data splits (Table 1).

For subtask A (Politeness Classification), the dataset includes a training set containing 4,895 instances and a validation set containing 693 instances. For subtask B (Category Prediction), additional training and validation files focusing on polite and impolite instances were provided, containing 2,049 and 291 instances, respectively.

Each training and validation file contains three columns: *Sentence*, which represents the input text of Arabic social media posts; *Source*, indicating the platform from which the text was collected (Companies, Shein, YouTube, or Tweet); and *label*, corresponding to the ground-truth annotation of the politeness level (Polite, Neutral, or Impolite).

For subtask B, the dataset provides eight additional columns that encode pragmatic annotations. These columns are grouped into four pairs, each consisting of a category label and its corresponding supporting keyword(s). The fields *criterion1–criterion4* indicate the assigned pragmatic categories, while *keywords1–keywords4* provide lexical evidence supporting each category assignment. This design enables multi-label annotation, where a single text may be associated with up to four pragmatic categories.

After the development phase, the test sets were released for final evaluation. The subtask A test

set contains 1,406 instances with two columns (*Sentence* and *Source*), with ground-truth labels withheld for blind evaluation via the leaderboard. The subtask B test set includes 588 instances with three columns (*Sentence*, *Source*, and overall *label*), while the fine-grained category labels were withheld to ensure blind evaluation.

The dataset reflects informal Arabic language usage across different social media platforms and includes modern standard Arabic and dialectal Arabic. The diversity of sources introduces variation in linguistic style, politeness strategies, and pragmatic expressions, making the task particularly challenging. Furthermore, linguistic phenomena found in social media posts such as slang usage, short texts, emoji use, and sarcasm pose additional challenges for accurate classification.

2.3. Task Challenges

Politeness detection in Arabic presents several challenges. First, Arabic exhibits rich morphological variation, which increases lexical diversity and complicates modelling. Second, social media texts often contain dialectal variations, spelling inconsistencies, and informal expressions. Third, politeness is a pragmatic phenomenon that depends on cultural norms and contextual interpretation, making it more difficult to capture using purely lexical features. Finally, some texts may express multiple pragmatic functions simultaneously, requiring models capable of handling multi-label classification.

2.4. Related Work

Politeness detection has been widely studied in computational linguistics, primarily focusing on English (Al-Khalifa et al., 2026a). Early work relied on handcrafted linguistic features, including politeness markers, lexical cues, and syntactic structures, to identify politeness strategies (Danescu-Niculescu-Mizil et al., 2013). These approaches were limited in their ability to generalise across domains.

Recent advances in pretrained language models have significantly improved performance across various NLP tasks. BERT (Devlin et al., 2019) and its variants have achieved state-of-the-art results in text classification, sentiment analysis, and pragmatic inference. These models learn contextualised representations that capture semantic and syntactic information, making them well-suited for politeness detection.

For Arabic, several pretrained transformer models have been developed to address the linguistic complexity and morphological richness of the language. AraBERT (Antoun et al., 2020) was one of the first pretrained BERT models specifically designed for Arabic. MARBERT (Abdul-Mageed et al., 2021) further improved performance by training on

¹<https://www.codabench.org/competitions/11955/>

Subtask	Split	Neutral	Polite	Impolite
A	Train	3,664 (74.85%)	815 (16.65%)	416 (8.50%)
	Valid	519 (74.89%)	115 (16.59%)	59 (8.51%)
	Total	4,183 (74.86%)	930 (16.64%)	475 (8.50%)
B	Train	-	1,338 (65.30%)	711 (34.70%)
	Valid	-	190 (65.29%)	101 (34.71%)
	Total	-	1,528 (65.30%)	812 (34.70%)

Table 1: Distribution of politeness labels across the training and validation splits.

large-scale Arabic social media data, making it particularly effective for informal and dialectal Arabic. CAMELBERT (Inoue et al., 2021) introduced a suite of pretrained models trained on diverse Arabic corpora, demonstrating strong performance across multiple downstream tasks.

Despite these advances, politeness detection in Arabic remains relatively under-explored due to the lack of large annotated datasets (Al-Khalifa et al., 2026a). This gap persists even in multilingual efforts, such as (Srinivasan and Choi, 2022), which constructed a large-scale politeness dataset across multiple languages but did not include Arabic. Consequently, prior Arabic NLP research has predominantly focused on related tasks such as sentiment analysis (Al Motairi and Hadwan, 2024), offensive language detection (Mubarak et al., 2021; Albalawi and Yafooz, 2025), and hate speech detection (Alhazmi et al., 2024).

One of the few studies addressing politeness classification in Arabic is (Al-Khalifa et al., 2024). The authors investigate the effectiveness of transfer learning and large language models (LLMs) for politeness classification in Arabic social media text. Using a manually annotated dataset of 500 tweets from X, they compare fine-tuned Arabic language models (MarBERT and CamelBERT) with LLMs (GPT-4o-mini, Cohere Command, and JAIS 30B Chat) under zero-shot and few-shot prompting settings. The findings indicate that LLMs, particularly JAIS, outperformed the fine-tuned models, achieving an F1-score of 70.87%. The study is further extended in (Al-Khalifa et al., 2026b), which introduces a larger dataset comprising 10,000 annotated samples and evaluates a broader range of models. Among these, MARBERT achieved the best overall performance, with an accuracy of 0.9119 and a macro-F1 score of 0.8582, outperforming all other evaluated models.

3. System Overview

3.1. Subtask A: Politeness Classification

We experimented with multiple transformer-based models, including multilingual models (XLM-R and XLM-R Large) and Arabic-specific models

(CAMELBERT, MARBERTv2, ARBERTv2, and GATE AraBERT v1). These models were fine-tuned individually on the provided training data to predict the politeness label.

To improve robustness and performance, we employed a weighted soft-voting ensemble that combines class-probability outputs of the individual models. Specifically, for an input text, each model produces a probability distribution over the politeness classes. Each distribution is multiplied by a predefined weight, and the weighted probabilities are summed across models. The final prediction is obtained by selecting the class with the highest combined probability.

The model weights were determined using a grid search on the validation set. Candidate weights ranging from 0.0 to 1.0 (with increments of 0.1) were evaluated under the constraint that their sum equals 1 (3,003 combinations). For each weight combination, we computed the macro-F1 score on the validation set, and the configuration achieving the highest macro-F1 score was selected for the final system.

The optimal weight configuration was: MARBERTv2 (0.20), CAMELBERT (0.10), AraBERTv2 (0.20), XLM-R (0.10), and XLM-R Large (0.40). The GATE AraBERT v1 model was assigned a weight of 0.0 and was therefore excluded from the final ensemble.

3.2. Subtask B: Category Prediction

Our approach consists of three main stages: data preprocessing and normalisation, MARBERT-based multi-label classification, and threshold-based post-processing.

3.2.1. Data Cleaning and Normalisation

We applied a comprehensive preprocessing and normalisation pipeline to improve data consistency and reduce noise in Arabic social media text. First, we removed Arabic diacritics and elongation characters (tatweel), which do not contribute to semantic meaning but increase lexical variability. We then normalised orthographic variants of Arabic letters to a canonical form, including mapping different

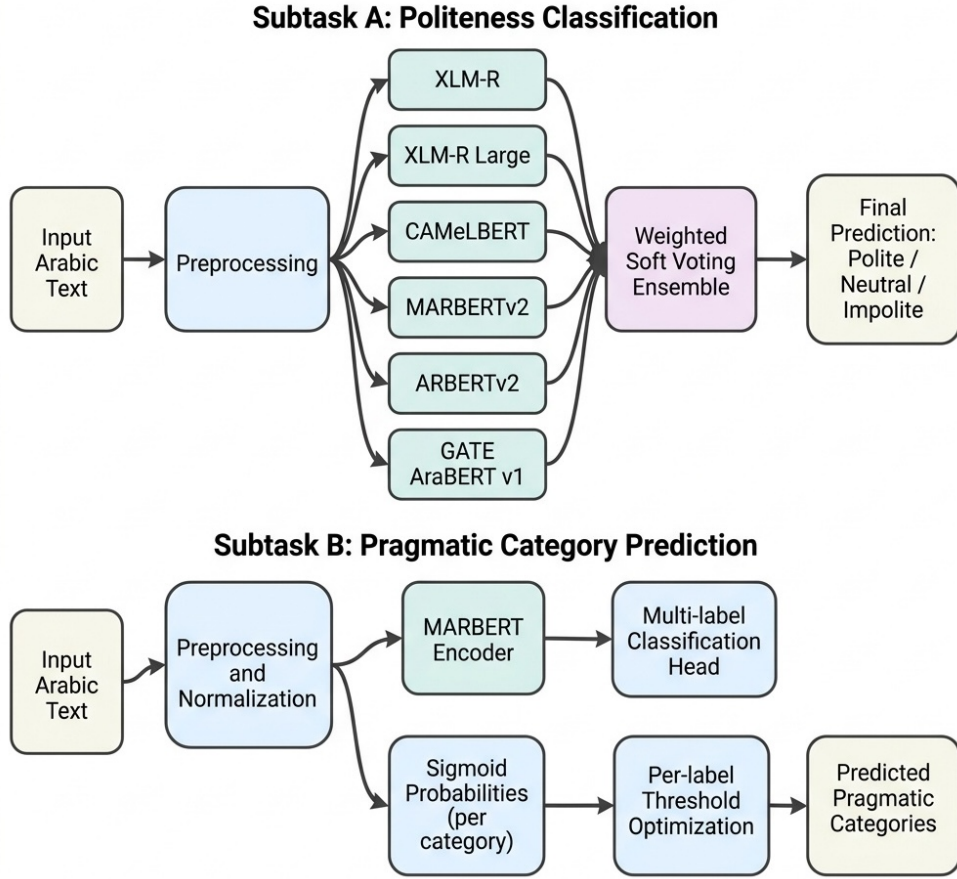


Figure 1: Overview of the proposed systems for the AdabEval 2026 shared task. The top pipeline shows the ensemble-based approach for politeness classification (Subtask A), while the bottom pipeline illustrates the MARBERT-based multi-label classification with threshold optimisation for pragmatic category prediction (Subtask B).

forms of Alef (آ, أ, إ) to a standard Alef (ا), converting Yeh variants (ي, ي) to (ي), normalising Waw variants (و, و) to (و), and converting Teh Marbuta (ة) to Heh (ه). These transformations reduce sparsity caused by orthographic variation.

To address informal writing and exaggerated expressions common in social media, we removed character elongation by collapsing repeated characters into a single instance (e.g., تكفونون → تكفون). We also removed URLs and user mentions, which do not contribute to pragmatic classification and may introduce noise. Finally, whitespace was normalised by removing extra spaces and ensuring consistent token separation.

In addition to text normalisation, we standardised pragmatic category annotations using a predefined mapping to canonical label names. The dataset contained variations in category labels due to differences in formatting, bilingual annotation (Arabic and English), and inconsistent spacing. We mapped all variants of each category to a sin-

gle normalised label (e.g., “Thanks & gratitude - الشكر و الامتنان” and “الشكر و الامتنان” were both mapped to *Thanks_and_gratitude*). When a direct mapping was unavailable, fallback normalisation rules were applied to extract the canonical English label and convert it into a consistent token format.

These preprocessing and normalisation steps improved label consistency, reduced vocabulary sparsity, and enhanced model performance by providing cleaner and more uniform input representations.

3.2.2. Model Architecture

Our system is based on MARBERT (Abdul-Mageed et al., 2021), a pretrained transformer model designed for Arabic social media text. MARBERT is built on the BERT architecture and trained on large-scale Arabic social media corpora, making it well suited for capturing informal language patterns and pragmatic cues. Given an input text x , MARBERT produces a contextualised representation, which is passed through a linear classification layer to gen-

erate logits for each category. Since subtask B is a multi-label classification task, we apply a sigmoid activation function independently to each category to produce probability scores.

3.2.3. Training and Threshold Optimisation

The MARBERT model (Abdul-Mageed et al., 2021) was fine-tuned on the provided training data using the HuggingFace Transformers library. During inference, the model outputs probability scores for each pragmatic category. To convert these probabilities into binary predictions, a decision threshold must be defined. Instead of using a fixed threshold (e.g., 0.5), we optimise label-specific thresholds on the validation set.

Specifically, for each category, we select the threshold that maximises its F1-score. This choice is motivated by the fact that threshold selection directly controls the trade-off between precision and recall, which significantly impacts classification performance (Lipton et al., 2014). Since the official evaluation metric is macro-F1 and the dataset is imbalanced across categories, optimising thresholds with respect to F1 is particularly appropriate (Sokolova and Lapalme, 2009).

Prior work has shown that tuning decision thresholds to maximise F1-score is effective in both binary and multi-label classification settings (Lipton et al., 2014). Therefore, we adopt per-label threshold optimisation to better capture category-specific characteristics.

3.2.4. Post-Processing and Prediction

At inference time, the model produces probability scores for each pragmatic category. We convert these scores into binary predictions using the optimised label-specific thresholds obtained from the validation set. A category is predicted if its probability exceeds the corresponding threshold.

This approach allows the model to account for differences in label distribution and classification difficulty across categories, particularly for low-frequency classes.

4. Experimental Setup

4.1. Data Splits

For subtask A, during the development phase, we augmented the training data by combining the training sets from subtask A and subtask B, as well as the validation set from subtask B. This strategy increased the amount of available training data and improved model generalisation. The validation set from subtask A was reserved exclusively for hyperparameter tuning, model selection, and ensemble weight optimisation.

For the final submission, we trained the models using the combined training and validation data from both subtask A and subtask B. The official subtask A test set, with hidden labels, was used for evaluation through the shared task leaderboard.

For subtask B, we followed the official dataset splits provided by the shared task organisers. The training set was used for model training, the validation set was used for hyperparameter tuning and threshold optimisation, and the test set was used for final evaluation.

4.2. Implementation Details

All experiments were implemented using the HuggingFace Transformers framework (Wolf et al., 2020) and PyTorch. We used pretrained transformer models that were fine-tuned on the provided dataset using GPU acceleration.

We used consistent training settings across all transformer models to ensure fair comparison and stable optimisation. The number of training epochs was determined separately for each model using early stopping based on validation performance.

All models were fine-tuned using the HuggingFace Transformers library with the AdamW optimiser and cosine learning rate scheduling. Mixed-precision training (FP16) was enabled when GPU resources were available to improve training efficiency. Tables 5 and 6 in the Appendix summarises the training hyperparameters used in our experiments.

4.3. Evaluation Metrics

The official evaluation metric used for leaderboard ranking is the macro-averaged F1-score. This metric assigns equal weight to each class or category, regardless of its frequency, making it particularly suitable for imbalanced datasets. Macro-averaged F1-score provides a balanced assessment of system performance across all politeness levels and pragmatic categories.

5. Results

Our systems achieved first place in both subtasks, demonstrating the effectiveness of transformer-based ensembling and MARBERT-based multi-label classification for Arabic politeness detection.

5.1. Subtask A: Politeness Classification

5.1.1. Validation Performance

Tables 2 and 3 present the validation performance of the individual transformer models and their ensemble. Among individual models, GATE AraBERT v1 achieved the highest macro-F1 score of 90.76%

and an accuracy of 85.59%. CAMeLBERT and ARBERTv2 also achieved strong performance, with macro-F1 scores exceeding 90%. The ensemble further improved overall performance by leveraging the complementary strengths of the individual models, resulting in improved robustness and generalisation compared to single-model approaches.

Model	Macro-F1	Accuracy
MARBERTv2	89.61	84.34
CAMeLBERT	90.91	85.37
XLM-R	88.74	80.21
XLM-R Large	88.46	81.84
ARBERTv2	90.04	84.92
GATE AraBERT v1	90.76	85.59
Ensemble	91.63	87.19

Table 2: Validation performance for subtask A.

Model	Neutral	Polite	Impolite
MARBERT	0.93	0.83	0.77
CAMeLBERT	0.94	0.86	0.76
XLM-R	0.93	0.82	0.65
XLM-R Large	0.92	0.84	0.70
ARBERT	0.93	0.84	0.78
GATE AraBERT v1	0.94	0.84	0.79
Ensemble	0.94	0.84	0.83

Table 3: Per-category validation F1-scores for subtask A.

5.2. Subtask B: Category Prediction

5.2.1. Validation Performance

We evaluated our MARBERT-based approach on the validation set to optimise classification thresholds. The optimal threshold was determined through a grid search to maximise macro-F1 score. We optimised decision thresholds separately for each category using the validation set. The optimal threshold for each category was selected to maximise its individual F1-score. This per-label threshold optimisation resulted in a macro-F1 score of 0.728 on the validation set.

Table 4 presents per-category F1-scores on the validation set. Performance was strongest on frequent categories such as gratitude, greetings, and prayers. The lower performance on categories such as hospitality was primarily due to limited training examples.

6. Conclusion

In this paper, we presented our systems for the AdabEval 2026 shared task on Arabic politeness detection and pragmatic category prediction. Our

Category	Per-label threshold	F1-score
Gratitude	0.10	0.898
Greetings	0.10	0.875
Prayers	0.10	0.839
Respect	0.10	0.794
Insult	0.30	0.789
Admiration / Love	0.10	0.716
Criticism	0.15	0.641
Hospitality	0.10	0.000

Table 4: Per-category validation F1-scores for subtask B.

approaches leveraged pretrained transformer models and ensemble learning to capture linguistic and pragmatic features in Arabic social media text.

For subtask A, we developed a weighted soft-voting ensemble combining multilingual and Arabic-specific transformer models. Our approach achieved a macro-F1 of 0.89 and 0.93 accuracy on the official test set, ranking first among all participants. The results demonstrate the effectiveness of ensemble learning in leveraging complementary strengths of diverse pretrained models.

For subtask B, we proposed a MARBERT-based multi-label classifier with validation-based threshold optimisation and post-processing. It achieved a macro-F1 of 0.58 on the official test set, also ranking first among all participants. The strong performance highlights the importance of domain-specific pretrained models and threshold optimisation for Arabic pragmatic category prediction.

Our results show that pretrained transformer models provide a robust framework for Arabic politeness detection, particularly when combined with ensemble techniques and threshold optimisation. However, performance on rare categories remains challenging due to limited training data.

In future work, we plan to explore data augmentation techniques, larger pretrained models, and instruction-tuned language models to further improve performance, particularly for low-resource pragmatic categories. We also aim to investigate more advanced multi-label learning approaches and prompt-based methods for pragmatic analysis.

7. Acknowledgments

We would like to thank the organisers of AdabEval 2026 for providing the datasets and organising this challenge.

8. Bibliographical References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, et al. 2021. Arbert & marbert: Deep

- bidirectional transformers for arabic. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing*, pages 7088–7105.
- Hend Al-Khalifa, Nadia Ghezaiel, and Maria Bounnit. 2024. Analyzing politeness in arabic tweets: A preliminary study. In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 352–359.
- Hend Al-Khalifa, Nadia Ghezaiel, Maria Bounnit, Hend Hamed Alhazmi, Noof Abdullah Alfear, Reem Fahad Alqifari, Ameera Masoud Almasoud, and Sharefah Ahmed Al-Ghamdi. 2026a. Adab: Arabic dataset for automated politeness benchmarking – a large-scale resource for computational sociopragmatics. In *Proceedings of the 2026 International Conference on Language Resources and Evaluation (LREC 2026)*, Palma, Mallorca, Spain.
- Hend Al-Khalifa, Nadia Ghezaiel, Maria Bounnit, Hend Hamed Alhazmi, Noof Abdullah Alfear, Reem Fahad Alqifari, Ameera Masoud Almasoud, and Sharefah Ahmed Al-Ghamdi. 2026b. Adab: Arabic dataset for automated politeness benchmarking—a large-scale resource for computational sociopragmatics. *arXiv preprint arXiv:2602.13870*.
- Reem K Al Motairi and Mohammed Hadwan. 2024. Sentiment analysis methods for arabic content on social media: A systematic review. *Ingénierie Des systèmes D’information*, 29(1).
- Amal Albalawi and WM Yafooz. 2025. Arabic offensive text classification using emojis: including emoji data in arabic natural language processing. *International Journal of Electrical and Computer Engineering*, 15(3):3332–3345.
- Ali Alhazmi, Rohana Mahmud, Norisma Idris, Mohamed Elhag Mohamed Abo, and Christopher Eke. 2024. A systematic literature review of hate speech identification on arabic twitter data: research challenges and future directions. *PeerJ Computer Science*, 10:e1966.
- Reem Alqifari, Hend Al-Khalifa, Nadia Ghezaiel Hammouda, Maria Bounnit, Hend Al-Hazmi, Ameera Almasoud, Sharefah AlGhamdi, and Noof Alfear. 2026. The adabeval 2026 shared task on arabic politeness detection. In *Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks (OSACT 2026)*, Palma, Mallorca, Spain. Co-located with the 2026 International Conference on Language Resources and Evaluation (LREC 2026).
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of the 4th workshop on open-source arabic corpora and processing tools, with a shared task on offensive language detection*, pages 9–15.
- Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 250–259.
- 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*.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.
- Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. 2014. Thresholding classifiers to maximize f1 score. *arXiv preprint arXiv:1402.1892*.
- Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2021. Arabic offensive language on twitter: Analysis and experiments. In *Proceedings of the sixth arabic natural language processing workshop*, pages 126–135.
- Priyanshu Priya, Mauajama Firdaus, and Asif Ekbal. 2024. Computational politeness in natural language processing: A survey. *ACM Computing Surveys*, 56(9):1–42.
- Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.
- Anirudh Srinivasan and Eunsol Choi. 2022. Tydip: A dataset for politeness classification in nine typologically diverse languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5723–5738.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical*

9. Appendix

9.1. Hyperparameter Configuration Details

Parameter	value
Max sequence length	256
Tokeniser	AutoTokeniser from pre-trained model
Data collator	DataCollatorWithPadding
Batch size (train)	8
Batch size (eval)	16
Gradient accumulation	2
Effective batch size	16
Learning rate	3×10^{-5}
Epochs	Up to 15 (early stopping)
Optimiser	AdamW
Weight decay	0.01
Warmup ratio	0.06
LR scheduler	Cosine
Mixed precision	FP16 if CUDA available
Threshold optimisation	Argmax (no threshold tuning)

Table 5: Configuration for subtask A.

Parameter	Value
Max sequence length	192
Tokeniser	AutoTokeniser from pre-trained model
Data collator	DataCollatorWithPadding
Batch size (train)	10
Batch size (eval)	10
Gradient accumulation	1
Effective batch size	10
Learning rate	2×10^{-5}
Epochs	Up to 50 (early stopping)
Optimiser	AdamW
Weight decay	0.01
Warmup ratio	0.06
LR scheduler	Linear (default Hugging-Face)
Mixed precision	FP16 if CUDA available
Threshold optimisation	Validation-based grid search

Table 6: Configuration for subtask B.

MOSKA-NLP at AdabEval 2026: Feature-Enriched Ensembling for Arabic Politeness Detection

Nina Andriyanova-Almaamary

College of Computer and Information Sciences, King Saud University
Riyadh, Saudi Arabia
ninamaamary@gmail.com

Abstract

In this paper, we present our system for subtask A of the AdabEval 2026 shared task, which focuses on classifying Arabic text into Polite, Neutral, and Impolite categories. Politeness detection is challenging because it cannot be inferred from lexical meaning alone. This is prominent in Arabic language, where politeness is often conveyed through formulaic expressions, stylistic cues, and dialectal variations. Our approach follows a three-stage strategy. First, we evaluate five Arabic sentence embedding models based on different pretrained encoders to identify a strong representation backbone. Second, we enrich sentence embeddings with explicit lexical, surface-level, and auxiliary signals derived from external models, including dialect, intent, and sarcasm classifiers. Third, we combine predictions from independently trained models, using weighted probability-level ensembling with class-specific decision thresholds to address class imbalance. Experimental results show that feature-enriched representations consistently outperform embedding-only baselines, with additional gains obtained from calibrated ensembling. The proposed system achieves a macro-F1 score of 0.87 and an accuracy of 93% on the official AdabEval 2026 evaluation for subtask A.

Keywords: Arabic NLP, Politeness Detection, Text Classification, Ensemble Learning

1. Introduction

Politeness detection is the task of classifying text according to how polite or impolite it is, typically using labels such as Polite, Neutral, and Impolite. Unlike tasks such as sentiment analysis, politeness cannot be determined from meaning alone. The same message can be polite or impolite depending on how it is phrased, which words are used, and how the person addresses others.

This challenge is prominent in the Arabic language. Politeness is often expressed through greetings, honorifics, religious expressions, and formulaic phrases (Ameri et al., 2023), and can vary across dialects and social context. Polite intent can be conveyed without positive sentiment, while impoliteness can rely on subtle phrasing rather than explicit insults.

The AdabEval 2026 shared task (Alqifari et al., 2026) provides a benchmark for Arabic politeness detection using the ADAB dataset (Al-Khalifa et al., 2026). Systems are evaluated primarily using macro-F1 score, while accuracy is reported as a complementary metric. The task is further complicated by severe class imbalance and indirect expressions of politeness; for example, sarcastic expressions can appear polite on the surface while conveying impoliteness in intent.

In contrast to approaches that rely mainly on end-to-end fine-tuning or LLM prompting, this work focuses on understanding how different combinations of sentence representations, explicit lexical and pragmatic features, and calibrated decision strategies affect performance on Arabic po-

liteness detection. Rather than proposing a new model, our goal is to identify which components contribute most under severe class imbalance and pragmatic ambiguity. Our experiments show that feature enrichment provides the largest gains over embedding-only baselines, while probability-level fusion with class-specific thresholds offers consistent improvements under imbalance. We also find that auxiliary signals such as sarcasm, dialect, and intent are most useful when used as complementary features rather than standalone predictors. On the official test set, the final system achieves a macro-F1 score of 0.87 and an accuracy of 93% ranking second in the shared task. Code is available at: <https://github.com/NinaM31/politeness-detection-arabic>.

2. Background

2.1. Dataset

We use the ADAB dataset released by AdabEval 2026: Arabic Politeness Detection (Subtask A). The official split contains 4,895 training instances, 693 validation instances, and 1,406 unlabeled test instances. The data includes both Modern Standard Arabic and multiple dialects (Gulf, Egyptian, Levantine, Maghrebi) and spans multiple domains, such as YouTube comments, product reviews, and twitter/X posts. The dataset is highly imbalanced, with Neutral instances accounting for approximately 75% of the data, while Polite and Impolite account for approximately 17% and 9% respectively, as shown in Table 1.

Split	Polite	Neutral	Impolite
Train	815 (17%)	3664 (75%)	416 (9%)
Valid	115 (17%)	519 (75%)	59 (9%)

Table 1: Label distribution across the official training and validation splits. Percentages are rounded to the nearest integer.

2.2. Task Setup

The task is formulated as a multi-class classification problem. Given a single Arabic sentence as input, the system assigns one of three labels: Polite, Neutral, or Impolite. Systems are evaluated primarily using macro-F1 score, with accuracy reported as a secondary metric.

Politeness is defined by how a statement is expressed rather than by its sentiment polarity; negative content can still be labeled as Polite if phrased respectfully. This makes the task particularly challenging, as it requires capturing pragmatic and stylistic cues beyond surface meaning. To illustrate these distinctions, we provide examples from the ADAB dataset along with English translations:

- **Polite:** Dissatisfaction expressed as a respectful request. Example:
صعب ولا يعمل بسهولة كل مره افتح التطبيق
يتعبني ارجو تطويره
(It is difficult and does not work easily every time I open the app it tires me, please improve it)
- **Neutral:** Dissatisfaction expressed without polite or aggressive cues. Example:
التحديث الاخير سيء جدا حيث لا يمكنني من
الدخول السريع أو الدخول ب الهاتف المصرفي
(The latest update is very bad, as it does not allow me to log in quickly or to log in using the banking phone)
- **Impolite:** Dissatisfaction expressed using insulting cues. Example:
تطبيق فاشل كان من افضل تطبيقات البنوك في
السابق لكن الان لا يرتقي بالخدمه للعملاء
(A failed app. It was among the best banking applications in the past, but now it does not rise to the level of customer service)

2.3. Related Work

Computational politeness research is traditionally grounded in linguistic frameworks such as [Brown and Levinson \(1987\)](#) politeness theory, which models politeness as a pragmatic phenomenon shaped by social context and speaker intent. Prior work has applied these ideas using surface-level cues such as hedging, indirect requests, and politeness markers ([Danescu-Niculescu-Mizil et al., 2013](#)).

Research on Arabic politeness detection is limited but has recently started to adopt modern transfer learning and large language models (LLMs). [Al-Khalifa et al. \(2024\)](#) investigate politeness analysis in Arabic social media text, comparing fine-tuned Arabic transformers such as MARBERT ([Abdul-Mageed et al., 2021](#)) and CamelBERT ([Inoue et al., 2021](#)) with zero-shot and few-shot LLM-based approaches. Their results demonstrate the potential of both paradigms, while highlighting challenges related to pragmatic knowledge, cultural context, emoji usage, and ambiguity in linguistic expression.

In general, Arabic text classification has been dominated by transformer-based models pretrained on a large-scale Arabic corpora. A systematic review by [Alammery \(2022\)](#) shows that MARBERT and AraBERT ([Antoun et al., 2020](#)) consistently perform strongly on informal and dialectal Arabic. Recent work on Arabic sentence embeddings, such as General Arabic Text Embedding ([Nacar et al., 2025](#)), further demonstrates that sentence transformers trained with contrastive objectives provide robust sentence-level representations for noisy and dialect-rich text, which is particularly important for capturing subtle pragmatic and stylistic distinctions in politeness expressions that are not always reflected in explicit lexical cues.

Our approach follows a three-stage strategy: (i) selecting a strong sentence representation backbone, (ii) enriching representations with explicit lexical and pragmatic features, and (iii) applying probability-level ensembling with class-specific thresholding to address severe class imbalance.

In contrast to prior work that relies primarily on end-to-end fine-tuning or LLM prompting, we combine feature enrichment with auxiliary Arabic-specific signals (dialect, intent, sarcasm) and calibrated thresholding to address pragmatic ambiguity and class imbalance. In particular, auxiliary models are used as feature sources rather than standalone predictors, and class-specific thresholds are applied to improve minority class behavior.

Rather than introducing new modeling components, our contribution lies in identifying which design choices are most effective for Arabic politeness detection and how they interact in practice.

3. System Overview

3.1. Sentence Embedding Backbone

We evaluate five sentence embedding models that differ in their underlying pretrained encoders and representation learning objectives. These include models based on MARBERT and AraBERT, as well as sentence transformers following the matryoshka representations learning paradigm introduced in Nacar et al. (2025). All sentence transformers are initially evaluated in a frozen setting, without task specific fine-tuning. Table 2 reports their validation macro-F1 scores.

Model	Macro-F1
Marbert-all-nli-triplet-Matryoshka(Matryoshka) ¹	0.797
MARBERTv2 ²	0.753
GATE-AraBert-v1 ³	0.714
Arabic-all-nli-triplet-Matryoshka ⁴	0.714
Arabic-Triplet-Matryoshka-V2 ⁵	0.707

Table 2: Frozen validation performance of candidate embedding backbones. Details in Appendix A

Based on frozen macro-F1 scores, Matryoshka and MARBERTv2 are selected as the strongest backbones. While Matryoshka achieves higher performance in the frozen setting, MARBERTv2 yields better results after fine-tuning and is therefore used as the primary backbone. Matryoshka is incorporated as a complementary signal within the ensemble. Detailed results of individual fine-tuned models are reported in Table 4.

3.2. Feature Enrichment

Manual lexical indicators. We construct manually curated lexicons of politeness-related markers by combining domain knowledge with inspection of the training data. Candidate expressions include greetings, honorifics, addressee markers, and common insulting terms. To ensure reliability, each keyword is validated on the training set and retained only if it exhibits high class specificity (> 0.75), defined as the proportion of occurrences of a keyword that appear in sentences of its most frequent label, divided by its total occurrences across all classes. The final lexicons contain approximately 30 Polite markers, 40 Impolite markers, 25 Addressee markers, and 25 Honorifics. Table 3 shows representative examples from the manually curated lexicons.

¹Marbert-all-nli-triplet-Matryoshka

²MARBERTv2

³GATE-AraBert-v1

⁴Arabic-all-nli-triplet-Matryoshka

⁵Arabic-Triplet-Matryoshka-V2

Category	Sample Markers
Polite terms	شكرا، من فضلك، يعطيك (thank you, please, give you)
Addressee markers	يا أخي، يا أختي، حضرتك (my brother, my sister, sir/ma'am)
Honorifics	أستاذ، دكتور، سعادة (professor, doctor, excellency)
Insult terms	فاشل، كذاب، زبالة (failure, liar, trash)

Table 3: Samples from manually curated lexical indicators used for politeness detection.

Automatically derived lexicons. We extract class-specific keywords from the training data by first removing globally frequent terms, appearing in more than 5% of documents, and then selecting words that appear mostly in a single class using frequency ratios. We retain only words that occur frequently (at least 10 times) and are strongly associated with class specificity (> 0.75). This results in dataset-adapted lexicons of size 29 Polite, 351 Neutral, and 5 Impolite.

Surface and auxiliary features. We incorporate lightweight surface and pragmatic cues, including character elongation, punctuation patterns, emoji counts, and diacritic-based signals. Lexical features are computed on both the original text and a normalized variant (alef/ya normalization, diacritic removal, and repetition reduction) to improve robustness to orthographic variation. We additionally include predictions from external pretrained models for dialect¹, intent², and sarcasm³ detection. These signals are used as complementary inputs rather than standalone predictions. As these models are not specifically evaluated on the ADAB dataset, their predictions may introduce noise due to domain differences.

3.3. Primary Classification Arm

After selecting the best-performing backbone and feature configuration, we construct a primary classification arm that serves as the anchor model. This arm uses fine-tuned MARBERTv2 embeddings combined with the selected features. On top of this representation, we train three linear classifiers: multinomial logistic regression (LR), a Stochastic Gradient Descent (SGD) classifier with log-loss, and Complement Naive Bayes (NB). LR and SGD operate on the full embedding + feature representation,

¹marbertv2-arabic-written-dialect-classifier

²Arabic-bank77-intent-classification

³sarcasm-classifier-bert-base-arabic-camelbert

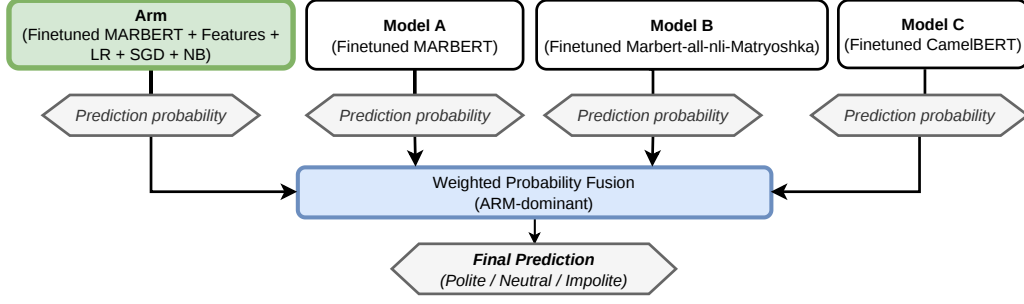


Figure 1: A primary classification arm based on fine-tuned MARBERT embeddings provides the dominant probability estimates, while independently trained auxiliary models contribute complementary signals.

while NB is trained on a sparse feature-only view. Their probability outputs are fused using weighted averaging, with weights 0.44 (LR), 0.46 (SGD), and 0.10 (NB), selected via grid search on the validation set. The SGD classifier is calibrated using sigmoid calibration with 3-fold cross-validation on the training set, while NB serves as a weak complementary signal. The resulting probabilities are then passed to class-specific thresholding to address class imbalance.

3.4. Auxiliary Models and Ensembling

To improve robustness, we incorporate auxiliary models trained with different pretrained backbones. Fine-tuned `MARBERTv2` and `Matryoshka` are included as top-performing models from our backbone experiments, while `CamelBERT`⁴ is added based on prior work (Al-Khalifa et al., 2024). These models are not used as standalone predictors but provide complementary probability estimates to the primary arm. Given an input x , each model outputs class probabilities, which are combined via weighted fusion:

$$\mathbf{p}(x) = \sum_{k \in \mathcal{K}} w_k \mathbf{p}^{(k)}(x), \quad \sum_{k \in \mathcal{K}} w_k = 1. \quad (1)$$

The primary arm is assigned a weight of 0.55, while auxiliary models receive smaller weights: 0.10 (`MARBERTv2`), 0.15 (`CamelBERT`), and 0.20 (`Matryoshka`), selected via grid search on the validation set. To address class imbalance, we use class-specific thresholds with Neutral as the default prediction. A prediction is overridden to Polite or Impolite only if the Neutral probability falls below its threshold and the most confident non-neutral class exceeds its corresponding class-specific threshold. The final thresholds are Neutral = 0.32, Polite = 0.40, and Impolite = 0.40, selected using grid search on the validation set. Figure 1 provides an overview of the system.

⁴CamelBERT

4. Experimental Setup

4.1. Hyperparameter Optimization

For frozen sentence embeddings, we train a multinomial logistic regression classifier and tune its hyperparameters through grid search over the regularization strength $C \in \{0.25, 0.5, 1, 2, 4, 8\}$ and solver choice (lbfgs, liblinear). NB is used with a fixed smoothing parameter $\alpha = 0.15$. For fine-tuned models and linear classifiers in the primary arm, hyperparameters are optimized using Optuna, we search over learning rate ($1e^{-5}$ – $5e^{-5}$), batch size $\{8, 16\}$, weight decay (0.01–0.1), number of epochs (2–3), and warmup ratio (0.05–0.15), using validation macro-F1 as the objective over 6 trials. For SGD classifiers, we tune regularization strength, penalty type, learning rate schedule, learning rate, maximum iterations, tolerance, and class weighting using Optuna with 30 trials and validation macro-F1 as the objective. The SGD classifier is calibrated using sigmoid calibration with 3-fold cross-validation on the training set. Experiments are implemented using scikit-learn (v1.6.1), Optuna (v2.10.1), and transformers (v4.57.1).

4.2. Feature Selection

Feature selection is performed through ablation experiments on the validation set. We evaluate 45 feature configurations in both frozen and fine-tuned settings. Across both settings, manually curated lexical features, character elongation, and sarcasm signals yield the largest gains, while dialect features provide complementary improvements. The best-performing configuration is used in the final system, with detailed ablation results provided in the Appendix B.

4.3. Ensembling and Thresholding

Final predictions are obtained through weighted probability fusion of the primary classification arm and auxiliary models. The primary arm is assigned the largest weight to preserve stability. Fusion

weights and class-specific thresholds are selected via grid search on the validation set and fixed for test-time evaluation. The final values are reported in Section 3. While this setup is standard in shared task settings, we note that optimizing multiple components on a single validation split may introduce some degree of overfitting to this data.

5. Results

Table 4 summarizes the progressive performance of our system. Starting from frozen sentence embeddings, each component contributes consistent improvements, including fine-tuning, feature enrichment, calibrated classification, and ensembling. The final system achieves a macro-F1 **0.87** and accuracy of **93%** on the official test set, ranking second in the shared task. Ablation experiments confirm that feature enrichment is crucial for Arabic politeness detection. Manually curated lexical features outperform automatically derived lexicons, particularly after normalization. Additionally, character elongation and sarcasm provided robust gains. Aggressively expanding class-specific lexical markers yields diminishing returns. Increasing impolite markers improve validation score up to 0.87 macro-F1, but reduces test performance to 0.86, indicating overfitting and increased cross-class overlap. These findings motivate the conservative feature selection strategy adopted in the final approach.

Stage	Split	Macro-F1	Acc. (%)
Frozen MARBERTv2	Valid	0.753	85
Frozen Matryoshka	Valid	0.797	87
+ Feature enrichment	Valid	0.845	90
Fine-tuned CamelBERT	Valid	0.822	90
Fine-tuned Matryoshka	Valid	0.834	90
Fine-tuned MARBERTv2	Valid	0.847	90
+ Feature enrichment	Valid	0.853	91
+ Arm (LR + SGD + NB)	Valid	0.859	92
Ensemble (development)	Valid	0.862	92
Ensemble (official)	Test	0.87	93

Table 4: Validation results are reported on the development set and official results on the test set. Feature enrichment represents the best configuration, combining manually curated lexical indicators, character elongation, and sarcasm signals. Detailed analysis is provided in Appendix C.

5.1. Error Analysis

We analyze misclassifications on the validation set to identify common error patterns. Most errors involve Impolite and Polite instances being misclassified as Neutral. This is reflected in Figure 2, where 36% of Impolite instances and 14% of Polite instances are predicted as Neutral, while Neutral is

correctly classified in 97% of cases. This indicates a strong tendency for the model to default to the Neutral class under ambiguity. A frequent error type involves Impolite instances expressed through evaluative language rather than explicit insults. For example, the sentence خدمة جدا سيئة وتحديث منها لا فائدة منه (“*Very bad service and the update has no benefit*”) is labeled as Impolite but predicted as Neutral, as it lacks clear insulting markers. Lexical ambiguity further contributes to confusion. The adjective سيء (“*bad*”) appears across both Neutral

and Impolite contexts, whereas terms such as فشل (“*failure*”) are consistently associated with impoliteness. These findings highlight the limitations of surface lexical cues and suggest the need for improved modeling of pragmatic context, such as whether a comment is directed at an addressee or expressed as a general statement. Additional error analysis examples are provided in Appendix D.

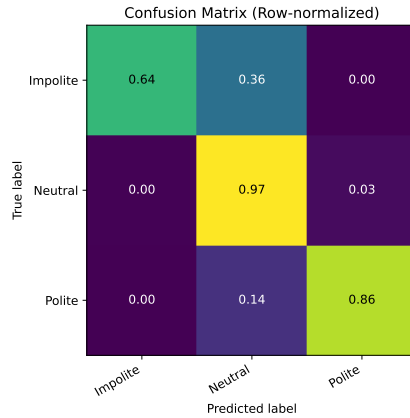


Figure 2: Normalized confusion matrix (validation).

6. Conclusion

We presented a feature-enriched and ensemble-based system for Arabic politeness detection. By combining sentence embeddings, explicit linguistic features, and calibrated decision strategies, the system achieves strong performance on the official test set. Our experiments provide practical insights: feature enrichment yields the largest gains, while auxiliary models are most effective as complementary signals rather than standalone predictors. We note that optimizing multiple components on a single validation split may introduce some overfitting, and reproducibility may depend on access to external pretrained models. The results also highlight the limitations of surface lexical cues, as lexical ambiguity often leads to confusion between classes. Future work includes modeling addressee awareness, improving contextual understanding, and better handling pragmatic ambiguity across dialects.

Acknowledgments

We thank the organizers of the AdabEval 2026 shared task for providing the dataset and timely support. We are also grateful to the anonymous reviewers for their helpful feedback, which improved the quality of this paper. Finally we thank the developers of the open-source pretrained models used in our experiments.

References

- Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. [ARBERT & MARBERT: Deep bidirectional transformers for Arabic](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.
- Hend Al-Khalifa, Nadia Ghezaiel, and Maria Bounnit. 2024. [Analyzing politeness in Arabic tweets: A preliminary study](#). In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 352–359, Trento. Association for Computational Linguistics.
- Hend Al-Khalifa, Nadia Ghezaiel, Maria Bounnit, Hend Hamed Alhazmi, Noof Abdullah Alfear, Reem Fahad Alqifari, Ameera Masoud Almasoud, and Sharefah Ahmed Al-Ghamdi. 2026. ADAB: Arabic dataset for automated politeness benchmarking – a large-scale resource for arabic politeness analysis. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).
- Ali Saleh Alammery. 2022. [Bert models for arabic text classification: A systematic review](#). *Applied Sciences*, 12(11).
- Reem Alqifari, Hend Al-Khalifa, Nadia Ghezaiel Hammouda, Maria Bounnit, Hend AlHazmi, Ameera Almasoud, Sharefah Al-Ghamdi, and Noof Alfear. 2026. The adabeval 2026 shared task on arabic politeness detection. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks*, Palma, Mallorca, Spain. European Language Resources Association (ELRA). Co-located with LREC 2026.
- Mohamadali Ameri, Ali Zeighami, and Sayyed Reza Mirahmadi. 2023. [A study of the polite method in the arabic language according to brown and levinson’s theory of politeness](#). *Studies on Arabic Language and Literature*, 13(36):61–88.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resources Association.
- Penelope Brown and Stephen C. Levinson. 1987. *Politeness: Some Universals in Language Usage*. Cambridge University Press.
- Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. [A computational approach to politeness with application to social factors](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259, Sofia, Bulgaria. Association for Computational Linguistics.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. [The interplay of variant, size, and task type in Arabic pre-trained language models](#). In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.
- Omer Nacar, Anis Koubaa, Serry Sibae, Yasser Al-Habashi, Adel Ammar, and Wadii Boulila. 2025. [Gate: General arabic text embedding for enhanced semantic textual similarity with matryoshka representation learning and hybrid loss training](#).

Appendix

A. Sentence Embedding Backbones

Table 5 reports the performance of evaluated sentence embedding models in the frozen setting.

Model	Macro-F1	C	Penalty	Solver
Matryoshka	0.797	0.50	l1	liblinear
MARBERTv2	0.753	2.00	l2	liblinear
GATE-AraBert-v1	0.714	0.25	l1	liblinear
A-all-Matryoshka	0.714	2.00	l2	liblinear
A-Matryoshka	0.707	0.25	l2	liblinear

Table 5: Macro-F1 scores for sentence embedding backbones using frozen embeddings and LR.

B. Feature Ablation Results

We conduct extensive ablation experiments to assess the contribution of lexical, surface, pragmatic, and auxiliary features. A total of 45 configurations are evaluated using a frozen encoder and repeated after fine-tuning. This section reports the full results for both settings and summarized in Appendix C.

Feature Groups.

- **Manual lexical (MNL)**: counts of polite phrases, honorifics, insults, and addressee markers.
- **Automatic lexical (Auto)**: count of class-specific keywords extracted from training data.
- **Clean (CLN)**: orthographically normalized variants of lexical features.
- **Surface (ELG)**: character elongation indicators.
- **Pragmatic (PRG)**: emoji count, repeated punctuation, exclamation and question marks, and diacritic statistics.
- **Source Data (SRC)**: text source provided by ADAB (Tweet, Companies, YouTube, Shein).

B.1. Frozen Matryoshka Encoder

This subsection reports ablation experiments using frozen Matryoshka sentence embeddings as the backbone. All feature groups are independently concatenated to the baseline embeddings only.

Main Experiment	Macro-F1	vs. Base (%)
Baseline (Base)	0.797	+0.00
+ SRC	0.796	-0.12
+ Auto	0.803	+0.75
+ SRC + Auto	0.811	+1.83
+ MNL	0.838	+5.20
+ SRC + MNL	0.841	+5.59

Table 6: Frozen Matryoshka encoder: main experiments obtained by independently concatenating the indicated feature sets to the baseline embedding vector. Improvements are reported relative to the baseline configuration (Base).

Pragmatic Experiment	Macro-F1	vs. Base (%)
Auto CLN + PRG	0.819	+2.80
Auto CLN + ELG + PRG	0.820	+2.91
MNL CLN + ELG + PRG	0.844	+5.89
MNL CLN + PRG	0.848	+6.38

Table 7: Frozen Matryoshka encoder: pragmatic feature ablation.

Preprocessing Experiment	Macro-F1	Base (%)
ELG	0.801	+0.52
Auto + ELG + SRC	0.812	+1.95
Auto + ELG	0.813	+2.03
Auto CLN	0.815	+2.33
Auto CLN + ELG	0.821	+3.02
MNL + ELG + SRC	0.838	+5.20
MNL + ELG	0.839	+5.32
MNL CLN + ELG	0.845	+6.07
MNL + Auto + ELG	0.846	+6.21
MNL CLN	0.846	+6.23

Table 8: Frozen Matryoshka encoder: preprocessing ablation results assessing the impact of surface elongation indicators (ELG) and orthographic normalization (CLN) on lexical feature extraction (MNL/Auto), optionally combined with the source metadata feature (SRC).

IDS Experiment	Macro-F1	vs. Base (%)
Auto + I	0.784	-1.55
I	0.785	-1.51
D	0.789	-0.97
S	0.797	+0.00
Auto + D	0.802	+0.68
Auto + S	0.803	+0.75
MNL + I	0.826	+3.63
MNL + Auto + I	0.834	+4.69
MNL + S	0.838	+5.20
MNL + D	0.841	+5.57
MNL + Auto + D	0.846	+6.14
MNL + Auto + S	0.846	+6.21

Table 9: Frozen Matryoshka encoder: ablation over auxiliary supervision signals for intent (I), dialect (D), and sarcasm (S), evaluated alone and in combination with lexical features (MNL/Auto).

Combined Experiment	Macro-F1	Base (%)
MNL + PRG + IDS	0.824	+3.46
MNL CLN + ELG + IDS	0.828	+3.90
MNL + IDS	0.828	+3.93
MNL + PRG + IDS + SRC	0.830	+4.15
MNL CLN + ELG + PRG + IDS	0.832	+4.42
MNL + PRG + S	0.837	+5.02
MNL + S	0.838	+5.20
MNL + DS	0.840	+5.39
MNL + PRG + DS	0.840	+5.39
MNL CLN + ELG + PRG + S	0.844	+5.89
MNL CLN + ELG + S	0.845	+6.07
MNL CLN + ELG + DS	0.845	+6.07
MNL CLN + ELG + PRG + DS	0.852	+6.90

Table 10: Frozen Matryoshka encoder: comprehensive feature combinations spanning lexical (MNL/Auto), normalization (CLN), surface elongation (ELG), pragmatic cues (PRG), auxiliary signals (IDS), and source metadata (SRC). The final row highlights the best frozen configuration (Best-M).

B.2. Fine-tuned MARBERT Encoder

This subsection reports the same ablation experiments after fine-tuning MARBERT on the task data. Results are reported relative to the best frozen Matryoshka configuration (Best-M) to facilitate direct comparison.

Main Experiment	Macro-F1	vs. Best-M (%)
Baseline (MARBERT)	0.845	-0.77
+ Auto	0.845	-0.77
+ SRC	0.844	-0.95
+ SRC + Auto	0.847	-0.59
+ MNL	0.849	-0.32
+ SRC + MNL	0.850	-0.14

Table 11: Fine-tuned MARBERT encoder: main experiments after encoder fine-tuning, where the same feature sets are independently concatenated to the fine-tuned baseline representation.

Pragmatic Experiment	Macro-F1	vs. Best-M (%)
MNL CLN + PRG	0.840	-1.38
Auto CLN + PRG	0.845	-0.77
MNL CLN + ELG + PRG	0.848	-0.49
Auto CLN + ELG + PRG	0.853	+0.12

Table 12: Fine-tuned MARBERT encoder: pragmatic ablation results measuring the contribution of PRG cues (emoji and punctuation statistics) when combined with normalized/elongation-aware lexical features.

Preprocessing Experiment	Macro-F1	Best-M (%)
Auto CLN	0.845	-0.77
ELG	0.849	-0.33
Auto + ELG	0.848	-0.43
Auto + ELG + SRC	0.849	-0.37
MNL CLN	0.849	-0.32
MNL + Auto + ELG	0.851	-0.10
Auto CLN + ELG	0.853	+0.12
MNL CLN + ELG	0.853	+0.12
MNL + ELG	0.853	+0.12
MNL + ELG + SRC	0.853	+0.12

Table 13: Fine-tuned MARBERT encoder: preprocessing ablation results evaluating normalization (CLN) and elongation indicators (ELG) under fine-tuning, optionally combined with lexical and source metadata features.

C. Experiment Analysis

Across frozen Matryoshka experiments, the lowest-performing configurations rely on auxiliary signals in isolation (e.g., Auto + I: 0.784), while the best performance is achieved by combining lexical, normalization, elongation, pragmatic, and auxiliary features (MNL CLN + ELG + PRG + DS: 0.852),

IDS Experiment	Macro-F1	vs. Best-M (%)
Auto + I	0.849	-0.32
I	0.846	-0.73
D	0.847	-0.55
S	0.845	-0.77
Auto + D	0.845	-0.77
Auto + S	0.845	-0.77
MNL + I	0.847	-0.59
MNL + Auto + I	0.847	-0.55
MNL + S	0.849	-0.32
MNL + D	0.849	-0.32
MNL + Auto + S	0.851	-0.10
MNL + Auto + D	0.853	+0.12

Table 14: Fine-tuned MARBERT encoder: ablation over auxiliary signals for intent (I), dialect (D), and sarcasm (S), evaluated alone and in combination with lexical features.

Combined Experiment	Macro-F1	Best-M (%)
MNL + PRG + DS	0.843	-1.05
MNL + PRG + IDS	0.843	-0.98
MNL CLN + ELG + PRG + IDS	0.844	-0.94
MNL + PRG + S	0.845	-0.77
MNL + PRG + IDS + SRC	0.846	-0.73
MNL CLN + ELG + IDS	0.847	-0.51
MNL CLN + ELG + PRG + DS	0.848	-0.49
MNL CLN + ELG + PRG + S	0.848	-0.49
MNL + IDS	0.849	-0.37
MNL + S	0.849	-0.32
MNL + DS	0.850	-0.14
MNL CLN + ELG + DS	0.853	+0.12
MNL CLN + ELG + S	0.853	+0.12

Table 15: Fine-tuned MARBERT encoder: comprehensive feature combinations spanning lexical, preprocessing (CLN/ELG), pragmatic (PRG), auxiliary (IDS/DS), and source metadata (SRC).

demonstrating that lexical features drive performance, with preprocessing and auxiliary signals providing gains only when integrated as complementary signals. In contrast, fine-tuned MARBERT experiments exhibit smaller performance differences, with the lowest configurations remaining close to the baseline (0.845), while the best combinations (MNL CLN + ELG + DS: 0.853) yield only modest gains. Notably, using the same feature combination (MNL CLN + ELG + DS), frozen Matryoshka achieves 0.845 compared to 0.853 with fine-tuned MARBERT, indicating that fine-tuning provides a consistent advantage while narrowing the gap between embedding backbones.

D. Additional Error Analysis

This section provides a qualitative analysis of representative misclassification cases observed on the validation set. We focus on frequent error types and illustrate them using full Arabic examples, with En-

glish translations, to highlight pragmatic and lexical challenges in Arabic politeness detection.

Impolite predicted as Neutral. The most common error pattern corresponds to Impolite instances misclassified as Neutral. These cases typically express strong dissatisfaction or criticism without explicit insults or abusive language. Negative sentiment is conveyed through evaluative adjectives rather than direct offense, leading the system to favor a Neutral prediction (true label: Impolite, predicted: Neutral):

خدمه سيئة حاولت انشط الحساب الدولي
واكلوني رصيد ع الفاضي وقالوا كلمة السر
غلط
"Bad service. I tried to activate the inter-
national account, they wasted my balance
for nothing and said the password was
wrong."

خدمة جدا سيئة وتحديث لا فائدة منه
"Very bad service and the update has no
benefit."

Although these sentences express clear dissatisfaction, they lack explicit insults or addressee markers, causing the model to interpret them as neutral complaints rather than impolite attacks.

Polite predicted as Neutral. Polite instances misclassified as Neutral often rely on implicit politeness strategies rather than explicit polite markers. These include praise, affective language, emojis, or indirect requests. In such cases, politeness is conveyed through tone and social context rather than formal expressions (true label: Polite, predicted: Neutral):

خدمه ممتازة واقتخر بأني من عملاءه
"Excellent service, and I am proud to be
one of its customers."

برنامج جميل جداً وسهل استخدام
"A very beautiful program and very easy
to use."

الدكتور هذا خبير في التغذية
"This doctor is an expert in nutrition."

Despite expressing positive affect or respect, the absence of explicit politeness markers causes the model to default to the Neutral class.

Neutral predicted as Polite. A smaller but notable error category involves Neutral instances predicted as Polite. These cases contain positive adjectives, encouragement, or softening expressions that resemble politeness markers, yet are annotated as Neutral in the dataset (true label: Neutral, predicted: Polite):

اتمنى الحلقة تترجم
"I hope this episode gets translated."

تكفون ابو مشاعل خلوه معكم الموسم كامل
معطي جو كبير للبودكاست
"Please, Abu Mishāal, keep him with you
for the whole season he adds great energy
to the podcast."

While such expressions may appear as polite request, they primarily function as opinions or positive feedback rather than explicit politeness acts. This suggests the presence of borderline or ambiguous annotation cases where pragmatic interpretation is subjective.

Lexical ambiguity in negative evaluative terms. Lexical ambiguity contributes substantially to misclassification. Some evaluative terms are strongly associated with impoliteness, while others appear across multiple classes. For example, the term فاشل ("failure") is consistently used as an impolite marker and is correctly classified in all observed cases. In contrast, the adjective سيء ("bad") frequently appears in neutral product or service complaints, leading to confusion between Neutral and Impolite (true label: Impolite, predicted: Neutral):

سيء جدا كل مره لازم ادخل رقم سري
للهاتف المصرفي وفي كل مره يقول خطأ
"Very bad. Every time I have to enter the
banking phone PIN, and every time it tells
me it's wrong."

This suggests that not all negative expressions carry the same pragmatic meaning, and that distinguishing between evaluative complaints and targeted impoliteness requires modeling contextual and pragmatic cues beyond surface-level sentiment.

SHU at AdabEval 2026: Category-Aware Fine-Tuning of MARBERT for Arabic Politeness and Pragmatic Function Classification

Alla Zwawi, Stephen Wu

Saraya Hamra University
Alsabaa Street, Tripoli, Libya
{a.zwawi, Dr.Stephen.wu}@shu.edu.ly

Abstract

This paper describes our submission to the AdabEval 2026 shared task, addressing Subtask A (politeness classification) and Subtask B (multi-label pragmatic category prediction). For Subtask A, we fine-tuned MARBERT using weighted cross-entropy to mitigate class imbalance. For Subtask B, we apply BCEWithLogitsLoss with inverse-frequency positive weighting to address the minority categories, and we introduce a category merging strategy to reduce categories' sparsity and annotation variation. Finally, we propose a stacked architecture where predicted pragmatic categories are injected as auxiliary features into the politeness classifier. Our results demonstrate that dialect-aware modelling, class-imbalance handling, and category-aware stacking improve Macro-F1 across both subtasks, achieving 0.85 for Subtask A and 0.55 for Subtask B on the test set.

Keywords: Politeness Detection, Transformer Encoder Fine-Tuning, Arabic NLP, MARBERT

1. Introduction

Politeness detection in Arabic text presents a challenge due to the language's various linguistic strategies to convey politeness rooted in cultural norms and social hierarchies (Ameri et al., 2023); social media text is particularly challenging due to dialectal diversity, orthographic variability, sarcasm, and culturally grounded expressions. Additionally, Arabic posts on social media often contain mixtures of Modern Standard Arabic (MSA) and regional dialects (Egyptian, Gulf, Levantine, and Maghrebi), along with informal spelling, elongations, and code-switching. Such characteristics present challenges in the automatic modelling of politeness and pragmatic functions. This shared task (Alqifari et al., 2026) addresses two related problems: Subtask A, which requires assigning one of three politeness labels (Polite, Neutral, Impolite) to each text sample, and Subtask B, a multi-label prediction of nine culturally grounded pragmatic categories (e.g., Criticism, Insult, Respect, Gratitude, Racism/Discrimination).

To motivate base-model selection, we conducted dialect identification using CAMEL Tools (Obeid et al., 2020) and exploratory data analysis to investigate orthography variation, code switching, and elongation. Our analysis revealed a substantial dialect content (over 70%) as well as mixed language text and elongation. Therefore, we selected MARBERT (Abdul-Mageed et al., 2021), a transformer model pretrained on large-scale Arabic Twitter data with broad dialect coverage. Since both subtasks are classification problems, we adopted an encoder-only architecture to enable efficient fine-tuning with lower computational cost.

Our approach combines dialect-aware modelling with class imbalance mitigation. For Subtask A, we apply weighted cross-entropy to address the dominance of the Neutral class. For Subtask B, we formulate the task as multi-label classification using BCEWithLogitsLoss with inverse-frequency positive weighting to improve minority category

recall. We further introduce a label-merging strategy to reduce annotation sparsity. Finally, we propose a category-aware stacking method in which predicted pragmatic category probabilities are incorporated into the politeness classifier.

Our experiments show that imbalance-adjusted training improves macro-level performance, that merging semantically overlapping criteria stabilises multi-label prediction, and that injecting pragmatic categories into politeness modelling reduces model confusion. Our approaches ranked 7th for Subtask A with a Macro-F1 of 0.85, and 2nd for Subtask B with a Macro-F1 of 0.55, evaluated on test data. Remaining challenges include sarcasm detection, which contributes to the model confusion between neutral and impolite classes, overlapping pragmatic categories such as Criticism and Insult, and calibration issues for underrepresented categories.

2. Background

The dataset is an annotated corpus consisting of Arabic social media posts from multiple sources (Shein, YouTube, X previously Twitter, etc). Each record contains an identifier, text, source, and primary label for subtask A (Polite, Impolite, Neutral), as well as up to four keyword-category pairs for subtask B. (Al-Khalifa et al., 2026). The input/output format is summarised in Table 1.

Input	Output Subtask A	Output Subtask B
لواء مفيد وفريد شكرا جزيلاً	Polite	Thanks & gratitude - الشكر و الامتنان

Table 1 Example of Input & Output

The dataset for subtask A consists of 4895 records for training, 693 records for evaluation, and 1406 records for testing. The data indicates a clear class imbalance, with Neutral class being the majority class, followed by Polite, while Impolite is the least represented. This motivates

us to use imbalance-mitigation training strategies to ensure improved recall for minority classes.

For Subtask B, the dataset contains 2,049 training records, 291 evaluation records, and 588 test records. The task involves predicting one or more of nine culturally grounded pragmatic categories: Criticism, Insult, Respect, Prayers, Greetings, Hospitality, Gratitude, Admiration / Love, and Racism / Discrimination. Subtask B also exhibits class imbalance, with categories such as Criticism and Respect occurring more frequently, while Hospitality and Racism / Discrimination are extremely rare. Furthermore, the dataset suffers from fragmentation and sparsity due to annotation criteria that are semantically overlapping or presented using different variations, which we addressed by introducing a category merging strategy in the preprocessing stage.

Al-Khalifa et al. (2024) presented a preliminary study on Arabic politeness detection with adequate performance using MARBERT as a base model. Their findings demonstrate the effectiveness of transformer-based transfer learning for modelling politeness in Arabic. However, their study focused on Modern Standard Arabic text and did not explicitly address the challenges posed by dialectal variation, multi-dialect mixing, or pragmatic multi-label classification in noisy social media environments. Our approach addresses these challenges in four ways:

- **Dialect-Aware Modelling:** Used MARBERT, guided by dialect analysis, for better coverage of dialectal social media text.
- **Robust Arabic Text Normalisation:** Applied a preprocessing pipeline for noisy dialectal content.
- **Label-Merging Strategy (Subtask B):** Merged semantically overlapping/variant criteria into nine official categories to reduce sparsity and annotation fragmentation.
- **Category-Aware Stacking (Subtask A):** Incorporated predicted pragmatic category probabilities from Subtask B into the Subtask A classifier, leveraging the pragmatics-politeness relation.

3. System Overview

Our system is based on fine-tuning MARBERT, a transformer encoder pretrained on a large set of Arabic tweets covering various dialects. We designed a specific model for subtask A and another for subtask B while sharing a unified preprocessing pipeline tailored to social media text noise.

3.1 Text Preprocessing

We implemented a text normalisation pipeline designed to reduce orthographic variation while

maintaining sentiment and pragmatic signals with the following steps:

- **URL Removal:** All hyperlinks (e.g., `http...`, `www...`) are removed.
- **Hashtag Normalisation:** Hashtag symbols (#) are removed while preserving the underlying word.
- **Tatweel Removal:** The elongation character (ـ) is removed.
- **Diacritics Removal:** Arabic diacritics are stripped to reduce noise.
- **Alef Normalisation:** Variants of Alef (آ, إ, ا) are normalised to (ا).
- **Ya Normalisation:** Alif Maqsurā (ي) is normalised to (ى).
- **Repeated Character Reduction:** Sequences of three or more repeated characters are reduced to at most two (e.g., `طوووول`, `طوول`).
- **Latin Letter Removal:** Non-Arabic letters are removed to reduce code-switching noise.
- **Emoji Preservation:** Emojis are retained, as they encode sentiment and pragmatic cues.
- **Whitespace Normalisation:** Multiple spaces, tabs, and line breaks are collapsed into a single space.

3.2 Subtask A Model

We fine-tune MARBERT¹ by attaching a linear classification head to the contextualised [CLS] token representation. To mitigate class imbalance, we compute inverse-frequency class weights and apply cross-entropy loss during training.

3.3 Subtask B Model

We fine-tune MARBERT with a sigmoid output layer producing independent probabilities for each category. Training uses BCEWithLogitsLoss with a positive class weighting term computed as:

$$weight_{pos} = \frac{count_{neg}}{count_{pos}}$$

This weighting increases the penalty for false negatives in rare classes, improving recall for infrequent categories such as Hospitality and Racism / Discrimination. At inference time, labels with probability ≥ 0.5 are selected; if none exceed the threshold, the highest-probability category is chosen, and if more than four are predicted, the top four are retained to comply with task constraints.

3.4 Category Merging Rules

For Subtask B, the dataset annotations contain semantically overlapping or non-official categories, which contribute to label sparsity and fragmentation. To reduce this issue, we introduce rule-based merging that maps related variants into the nine official categories. Asking for Permission is mapped to Respect; Threat and Verbal Violence are mapped to Insult; Sarcasm and Accusation variants are mapped to Criticism; and Sectarian or discrimination-related variants

¹ <https://huggingface.co/UBC-NLP/MARBERT>

are mapped to Racism / Discrimination. This consolidation reduces sparsity and improves label consistency.

3.5 Category-Aware Stacked Model

To utilise the relationship between pragmatic function and politeness, we incorporate Subtask B predictions into Subtask A through a stacked architecture. After training the multi-label subtask B model, we obtain the category probability vector $p_B \in \mathbb{R}^9$ from the model’s predicted output distribution. Specifically, for each instance in the Subtask A training data we generate p_B , by applying Subtask B model in inference mode, and concatenate it with the MARBERT [CLS] embedding h used for politeness classification as follows:

$$z = [h \oplus p_B]$$

The combined representation is passed through a projection and classification layer to produce the final politeness prediction. This integration allows pragmatic signals, such as Insult and Gratitude, to directly inform politeness modelling, in aim to reduce confusion between classes.

4. Experimental Setup

We use the official train/eval/test splits provided by the shared task organisers. We use the training and evaluation sets for model training and hyperparameter validation, while final predictions are generated on the held-out test sets.

All experiments use the preprocessing pipeline described in Section 3.1. Tokenisation is performed using the MARBERT tokeniser with a maximum sequence length of 128 tokens, and Inputs longer than this limit are truncated.

Models are fine-tuned using the AdamW optimiser with a learning rate of $2e-5$, a batch size of 16, and 3 epochs of training. Mixed-precision (FP16) training is enabled to improve computational efficiency. For Subtask A, we use weighted cross-entropy loss to address class imbalance. For Subtask B, we use BCEWithLogitsLoss with positive class weights ($weight_{pos}$).

Our implementation uses the Hugging Face Transformers library (v4.x)², PyTorch (v2.x.)³ and Datasets (v2.x)⁴. Dialect analysis is performed using CAMEL Tools (Obeid et al., 2020)⁵.

For Subtask A, we evaluate Accuracy, Macro-Precision, Macro-Recall, and Macro-F1, with Macro-F1 serving as the primary evaluation metric due to class imbalance. For Subtask B, we evaluate Micro-F1, Macro-F1, Macro-Precision, and Macro-Recall. Macro-F1 is emphasised to ensure balanced evaluation across rare and frequent categories. Macro-F1 score on the evaluation set is used for model selection, with the

highest-scoring checkpoint chosen as the final submission for official ranking in both subtasks.

5. Results & Discussion

5.1 Quantitative Results

This section details the results of our experiments; Table 2 illustrates the performance of different configurations for Subtask A, evaluated on the validation set. We compare a baseline MARBERT model, a version incorporating text preprocessing and weighted cross-entropy loss, and a category-aware stacked model that integrates Subtask B predictions, with preprocessing and weighted cross-entropy loss.

Model	Accuracy	Macro-Precision	Macro-Recall	Macro-F1
MARBERT (Baseline)	0.91	0.852	0.83	0.839
MARBERT +Preprocess +CrossEntropy	0.91	0.854	0.838	0.843
MARBERT Category-Aware	0.913	0.856	0.827	0.839

Table 2 Subtask A Performance Evaluation

The baseline MARBERT model already achieves strong performance due to its pretraining on Arabic social media data. Incorporating the preprocessing pipeline and class-weighted cross-entropy improves Macro-F1, primarily by increasing recall for minority classes. These improvements confirm that normalisation tailored to dialectal Arabic text reduces orthographic noise and improves model generalisation.

The category-aware stacked model achieves the highest accuracy (0.913) and macro-precision (0.856). However, its Macro-F1 remains comparable to the baseline because the additional pragmatic features slightly reduce macro-recall. This suggests that pragmatic signals may improve precision in Arabic politeness detection but may also lead to confusion in borderline cases.

Model	Micro-F1	Macro-Precision	Macro-Recall	Macro-F1
MARBERT +BCEWithLogitsLoss (Baseline)	0.773	0.669	0.531	0.58

² <https://huggingface.co/transformers>

³ <https://pytorch.org>

⁴ <https://huggingface.co/docs/datasets>

⁵ <https://camel-tools.readthedocs.io>

MARBERT +BCEWith LogitsLoss +Criteria Merging	0.784	0.669	0.922	0.746

Table 3 Subtask B Performance Evaluation

Table 3 presents the evaluation results for Subtask B, comparing the baseline MARBERT multi-label classifier, fine-tuned with BCEWithLogitsLoss, and the configuration incorporating a criteria-merging strategy applied as a preprocessing step. This strategy consolidates semantically overlapping labels prior to model training. The baseline model achieves a Micro-F1 of 0.773 but a relatively low Macro-F1 of 0.580, reflecting the strong class imbalance in the dataset, where frequent categories dominate the micro-averaged metric while rare categories remain poorly predicted.

Introducing the criteria-merging strategy significantly improves Macro-F1, mainly due to an increase in macro-recall. The criteria-merging strategy increases effective training instances for sparse labels and reduces fragmentation, advancing the model’s ability to generalise across pragmatically related expressions. The Micro-F1 also increases slightly, indicating that the merging strategy improves performance without sacrificing accuracy on frequent categories.

5.2 Error Analysis

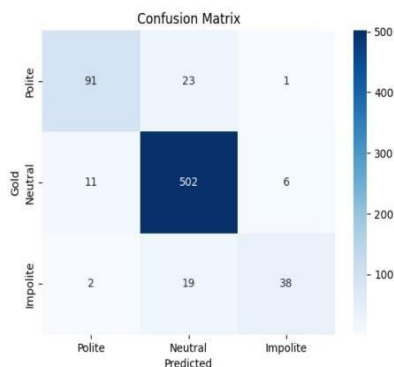


Figure 1 Confusion Matrix for Subtask A

For Subtask A, we analyse model errors using the confusion matrix shown in Figure 1 and qualitative analysis. The model performs best on the Neutral class, reflecting the model’s bias toward the majority class. The Polite class is predicted reliably, although 20% Polite instances are misclassified as Neutral when explicit politeness markers (e.g., gratitude, greetings) are absent. The most challenging class is Impolite, where only 64% of instances are classified correctly, compared to 79% for Polite and 97% for Neutral. 32% of impolite instances are misclassified as Neutral which suggests that negative or critical expressions are often interpreted as neutral when explicit insult cues are not present.

Qualitative inspection reveals recurring error types, supported by quantitative trends in misclassified Impolite instances. Among these errors, 35% contain evaluative words such as “سيء” (bad), where negative judgment is expressed without explicit offense, leading to Neutral predictions. Sarcasm accounts for 25% of errors, reflecting the model’s reliance on lexical cues rather than intended meaning. Similarly, 20% involve religious expressions (e.g., “والله ان” “تريبتكم صحيحه”) used to convey indirect criticism, which are often interpreted literally. These patterns align with confusion matrix results, where Impolite instances are frequently misclassified as Neutral, indicating systematic difficulty in capturing implicit negativity. Notably, many misclassified examples have high prediction confidence (90% or higher), indicating the model is confidently incorrect when pragmatic meaning differs from lexical sentiment.

Error analysis for Subtask B shows that errors are concentrated in rare categories, particularly Hospitality and Racism/Discrimination (support ≈ 1), reflecting severe class imbalance which leads the model to default to more frequent categories, such as Criticism or Respect, instead. Additionally, semantic overlap between pragmatic categories (e.g., admiration, gratitude, respect) which appear in similar linguistic contexts leads to label confusion. These results highlight class imbalance and pragmatic overlap as key challenges in multi-label politeness classification for Arabic social media posts.

6. Conclusion

This paper presented a system for Arabic politeness detection and pragmatic category prediction in social media text. We fine-tuned MARBERT for both subtasks and introduced improvements, including Arabic text normalisation, class-imbalance handling, a label-merging strategy for Subtask B, and a category-aware stacked model integrating pragmatic signals into politeness classification. Results show that MARBERT performs strongly across subtasks; while preprocessing and label consolidation improve robustness under dialectal variation and sparse annotations. Despite these contributions, several limitations remain. The approach relies on predicted pragmatic labels, which may propagate errors in the stacked model. Performance is also constrained by severe class imbalance and limited support for rare categories, and difficulty capturing implicit meaning such as sarcasm and indirect criticism. Future work should focus on improving robustness to implicit pragmatic cues, particularly sarcasm, and exploring more reliable integration of pragmatic signals.

7. Bibliographic References

- Ameri, M., Zeighami, A., & Mirahmadi, S. R. (2023). A study of the polite method in the Arabic language according to Brown and Levinson's theory of politeness. *Studies on Arabic Language and Literature*, 13 (36), 61–88. <https://doi.org/10.22075/lasem.2022.23600.1286>
- Hend Al-Khalifa, Nadia Ghezaiel, and Maria Bounnit. 2024. Analyzing Politeness in Arabic Tweets: A Preliminary Study. In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 352–359, Trento. Association for Computational Linguistics.
- Hend Al-Khalifa, Nadia Ghezaiel, Maria Bounnit, Hend Hamed Alhazmi, Noof Abdullah Alfear, Reem Fahad Alqifari, Ameera Masoud Almasoud and Sharefah Ahmed Al-Ghamdi. ADAB: Arabic Dataset for Automated Politeness Benchmarking - A Large-Scale Resource for Computational Sociopragmatics. 2026. In *Proceedings of The 2026 International Conference on Language Resources and Evaluation (LREC2026)*, Palma, Mallorca (Spain).
- Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.
- Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMEL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.
- Reem Alqifari, Hend Al-Khalifa, Nadia GHEZAIEL HAMMOUDA, Maria BOUNNIT, Hend AlHazmi, Ameera Almasoud, Sharefah AlGhamdi and Noof Alfear. 2026. The AdabEval 2026 shared task on Arabic Politeness Detection. In *Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks (OSACT 2026)* co-located with the 2026 International Conference on Language Resources and Evaluation (LREC2026), Palma, Mallorca (Spain).

Comparative Study of Machine Learning and Transformer-Based Approaches for Arabic Politeness Detection at AdabEval 2026

Mariam Ben Arbia¹, Ghada Ben Amor¹, Housseem Safi², Omar Trigui^{1,2}

¹The Higher Institute of Management of Sousse (ISGS), University of Sousse, Tunisia

²ANLP Research Group, MIRACL Laboratory, Sfax, Tunisia

Abstract

This paper describes our system submitted to the OSACT7 AdabEval shared task on Arabic politeness detection (Task A). The task requires classifying Arabic texts into three categories: *Polite*, *Impolite*, and *Neutral*. We systematically explore multiple approaches, progressing from classical machine learning baselines using pre-trained embeddings to fine-tuned transformer models. Our best system leverages MARBERT, a transformer model pre-trained on one billion Arabic tweets, fine-tuned with Focal Loss to handle the significant class imbalance present in the dataset (~75% *Neutral*). We additionally experiment with hybrid approaches combining fine-tuned embeddings with gradient-boosted classifiers and ensemble methods. Our best single model achieves a macro F1 score of 0.84 and an accuracy of 0.90 on the validation set, substantially outperforming classical ML baselines (F1 = 0.42).

Keywords: Arabic NLP, politeness detection, MARBERT, Focal Loss, text classification, shared task

1. Introduction

Politeness detection in natural language is an important task for understanding social dynamics in on-line communication (Danescu-Niculescu-Mizil et al., 2013). While significant progress has been made for English, Arabic politeness detection remains underexplored despite the language’s rich pragmatic features and wide dialectal variation (Brown and Levinson, 1987).

The OSACT7 AdabEval shared task (Alqifari et al., 2026) addresses this gap by proposing a three-way classification task: given an Arabic text (tweet, product review, or YouTube comment), the system must predict whether the text is *Polite*, *Impolite*, or *Neutral*. This task presents several challenges:

- **Class imbalance:** The dataset is heavily skewed toward the *Neutral* class (~75%), with *Polite* representing ~17% and *Impolite* only ~8.5%.
- **Dialectal diversity:** The data includes Modern Standard Arabic (MSA) as well as multiple Arabic dialects (Egyptian, Saudi, Gulf, etc.).
- **Implicit politeness:** Politeness cues in Arabic can be subtle and context-dependent, involving honorifics, religious expressions, and indirect speech acts.

In this paper, we describe our systematic exploration of multiple approaches, from feature-based classical machine learning to fine-tuned transformers. We show that the choice of pre-trained model is critical: MARBERT (Abdul-Mageed et al., 2021), trained on Arabic tweets, dramatically outperforms AraBERT (Antoun et al., 2020), which was primarily

trained on formal Arabic text. We further demonstrate that Focal Loss (Lin et al., 2017) is more effective than standard cross-entropy for handling the class imbalance inherent in this task.

2. Related Work

Computational approaches to politeness detection were pioneered by Danescu-Niculescu-Mizil et al. (2013), who developed a framework for English based on linguistic features derived from politeness theory (Brown and Levinson, 1987). For Arabic, the field is comparatively nascent, with most work focusing on sentiment analysis rather than politeness per se.

Pre-trained language models have become the dominant approach for Arabic NLP tasks. AraBERT (Antoun et al., 2020) was trained on news articles and Wikipedia, making it well-suited for MSA but less effective on dialectal text. MARBERT (Abdul-Mageed et al., 2021) addresses this limitation by pre-training on one billion Arabic tweets, capturing dialectal patterns and informal language use. The effectiveness of domain-matched pre-training for downstream tasks has been widely documented in the literature.

Focal Loss (Lin et al., 2017), originally proposed for object detection, has been successfully applied to text classification tasks with imbalanced data. By down-weighting well-classified examples, it allows the model to focus on hard, ambiguous cases — a particularly relevant property for politeness detection where the boundaries between classes can be subtle.

3. Data

The AdabEval dataset for Task A (Al-Khalifa et al., 2026) consists of Arabic texts from three sources: Twitter, company reviews, and Shein product reviews. Each text is labeled as *Polite*, *Impolite*, or *Neutral*.

Split	Total	Neut.	Pol.	Imp.
Train	4,895	74.9%	16.6%	8.5%
Val	693	74.9%	16.6%	8.5%
Test	1,406	—	—	—

Table 1: Dataset statistics for Task A. The test set labels were not released during the competition.

As shown in Table 1 and Figure 1, the dataset exhibits a strong class imbalance, with the *Neutral* class representing approximately 75% of the data. This imbalance motivates our use of specialized loss functions and resampling techniques.

4. Methodology

We explore three families of approaches, each building upon insights from the previous one. Figure 2 illustrates the architecture of our best-performing system.

4.1. Text Preprocessing

All approaches share a common preprocessing pipeline:

1. Removal of URLs, user mentions (@), and hashtags (#).
2. Arabic character normalization: all variant forms of *alef* are unified to bare *alef*, *ta marbuta* is mapped to *ha*, and *alef maqsur* to *ya*.
3. Whitespace normalization.

For AraBERT-based experiments, we additionally apply the AraBERT preprocessor, which performs segmentation and further normalization steps specific to that model.

4.2. Approach 1: Feature Extraction with Classical ML

Our first approach uses pre-trained AraBERT v2 as a fixed feature extractor. We extract [CLS] token embeddings (768 dimensions) from the model without any fine-tuning. These embeddings are then used to train several classical ML classifiers:

- Logistic Regression
- Random Forest

- XGBoost (Chen and Guestrin, 2016)
- Support Vector Machine (SVM)

This approach serves as our baseline, establishing the performance achievable without task-specific model adaptation.

4.3. Approach 2: Fine-Tuning with Cross-Entropy

Building on the baseline results, we fine-tune AraBERT v2 end-to-end with a classification head and standard weighted cross-entropy loss. We experiment with multiple learning rates and training configurations.

4.4. Approach 3: MARBERT with Focal Loss

Our best approach makes two key modifications:

Model selection. We replace AraBERT with MARBERT (Abdul-Mageed et al., 2021), which was pre-trained on one billion Arabic tweets. Given that our dataset includes tweets and informal reviews, this model provides a better domain match.

Loss function. We replace weighted cross-entropy with Focal Loss (Lin et al., 2017), defined as:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (1)$$

where p_t is the predicted probability for the correct class, $\gamma = 2.0$ is the focusing parameter that down-weights easy examples (this value follows the recommendation of the original paper (Lin et al., 2017)), and α_t represents per-class weights computed inversely proportional to class frequency.

Training details. We fine-tune MARBERT with the following hyperparameters. The learning rate and weight decay follow standard recommendations for BERT-style fine-tuning; the batch size and sequence length were chosen to fit GPU memory constraints; early stopping patience was set to 4 based on preliminary development set experiments:

- Learning rate: 1×10^{-5} with linear warmup (10% of total steps)
- Optimizer: AdamW (Loshchilov and Hutter, 2019) with weight decay 0.01
- Batch size: 16
- Maximum sequence length: 128 tokens
- Gradient clipping: max norm 1.0
- Early stopping: patience of 4 epochs
- Maximum epochs: 15

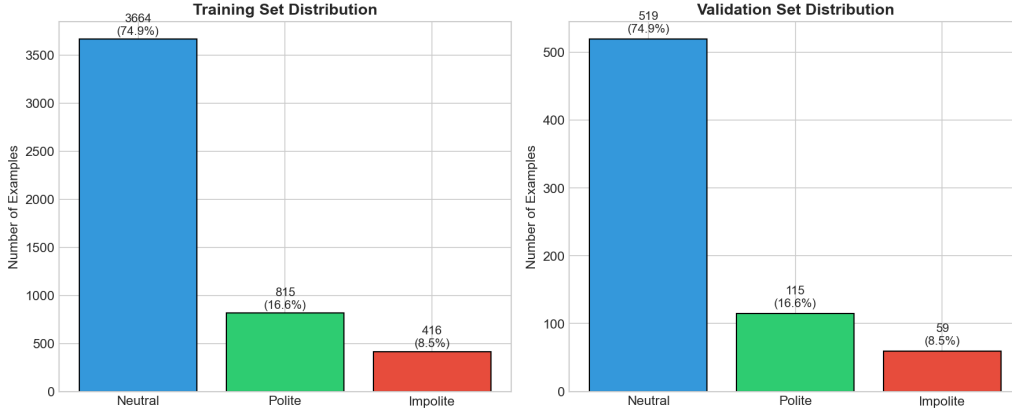


Figure 1: Class distribution in the training and validation sets. The *Neutral* class dominates both splits, representing approximately 75% of the data.

4.5. Approach 4: Hybrid (Fine-Tuned Embeddings + XGBoost)

We also explore a hybrid approach: extracting [CLS] embeddings from the *fine-tuned* MARBERT model and using them as features for an XGBoost classifier (Chen and Guestrin, 2016). To address class imbalance at the feature level, we apply SMOTE (Chawla et al., 2002) to the training embeddings before training XGBoost.

4.6. Approach 5: Ensemble

Finally, we experiment with soft-voting ensembles combining predictions from multiple approaches. We test two ensemble configurations:

- **All:** Combining all four individual models.
- **FT + Hybrid:** Combining the best fine-tuned model with the best hybrid model.

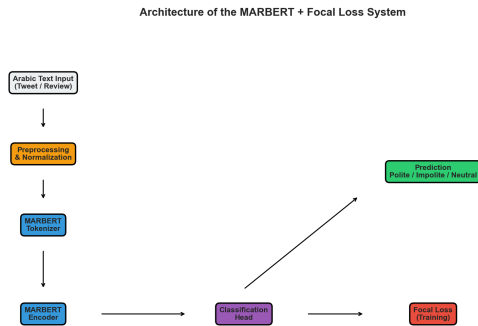


Figure 2: Architecture of our best system: MARBERT fine-tuned with Focal Loss for three-way politeness classification.

5. Results

5.1. Comparison of Approaches

Table 2 presents the results of all approaches on the validation set. The comparison is also visualized in Figure 3.

Approach	Acc	P	R	F1
<i>Classical ML</i>				
AraBERT+XGB	.69	—	—	.42
<i>Fine-tuning</i>				
AraBERT v2	.30	.40	.38	.27
MARBERT+FL	.90	.82	.88	.84
<i>Hybrid</i>				
MARB.→XGB	.68	.32	.33	.32
AraB.→XGB	.49	.33	.33	.32
<i>Ensembles</i>				
All models	.83	.81	.57	.64
FT+Hybrid	.86	.76	.74	.75

Table 2: Results on the validation set. P = Macro Precision, R = Macro Recall, F1 = Macro F1. FL = Focal Loss, XGB = XGBoost+SMOTE. For Classical ML, all four classifiers (LR, RF, SVM, XGBoost) were evaluated; XGBoost achieved the best macro F1 (0.42) and is reported here as the representative baseline.

The results reveal several important findings:

MARBERT vs. AraBERT. MARBERT fine-tuning achieves F1 = 0.84, while AraBERT v2 fine-tuning yields only F1 = 0.27. This dramatic difference highlights the importance of domain-matched pre-training: MARBERT, trained on Arabic tweets, is far better suited for informal text classification than AraBERT, trained primarily on news and Wikipedia.

Fine-tuning vs. Feature Extraction. The classical ML baseline using frozen AraBERT embed-

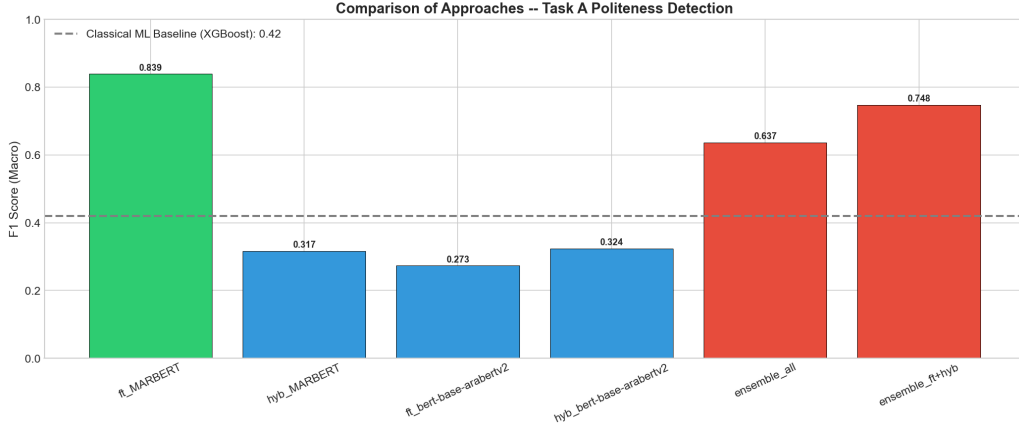


Figure 3: Macro F1 scores across all approaches on the validation set. Bars are colour-coded: green = best fine-tuned model (MARBERT + Focal Loss); red = ensemble variants; blue = all other approaches (fine-tuning baselines, hybrid models). The dashed grey line marks the classical ML baseline (F1 = 0.42).

dings (F1 = 0.42) actually outperforms AraBERT fine-tuning (F1 = 0.27), suggesting that fine-tuning AraBERT on this small, domain-mismatched dataset leads to catastrophic forgetting or poor convergence.

Focal Loss. MARBERT with Focal Loss achieves balanced performance across all three classes, with the focusing mechanism ($\gamma = 2.0$) effectively preventing the model from being dominated by the majority *Neutral* class. While we did not conduct a direct comparison of AraBERT with Focal Loss (which would isolate the effect of the loss function from the model choice), the large gap between AraBERT fine-tuning (F1 = 0.27) and MARBERT fine-tuning (F1 = 0.84) is primarily attributable to domain mismatch rather than the loss function. The choice of MARBERT over AraBERT is motivated by its pre-training on Arabic social media data, and the choice over other Arabic BERT-based models by its demonstrated state-of-the-art performance on dialectal Arabic tasks (Abdul-Mageed et al., 2021).

Hybrid approaches. Extracting embeddings from fine-tuned models and training XGBoost + SMOTE (F1 = 0.32) performs worse than direct fine-tuning, likely because the dimensionality reduction loses task-relevant information captured in the attention layers.

Ensembles. Ensemble methods do not improve upon the best single model, as the weaker models introduce noise that degrades overall performance.

5.2. Per-Class Analysis

Table 3 shows the detailed per-class results for our best model (MARBERT + Focal Loss).

The model achieves strong performance across all three classes. The *Neutral* class benefits from its larger representation (F1 = 0.94), and the minor-

Class	Prec.	Rec.	F1	Sup.
Impolite	.70	.81	.75	59
Neutral	.96	.91	.94	519
Polite	.79	.90	.85	115
Macro Avg	.82	.88	.84	693

Table 3: Per-class results of the best model (MARBERT + Focal Loss) on the validation set.

ity classes achieve F1 scores of 0.85 for *Polite* and 0.75 for *Impolite*. The *Impolite* class is the most challenging, which is expected given both the diversity of expressions of impoliteness across Arabic dialects and the fact that it is the most underrepresented class ($\sim 8.5\%$ of training data).

Figure 4 shows the confusion matrix for our best model. The main error patterns are: 20 *Neutral* examples misclassified as *Impolite*, 25 *Neutral* examples misclassified as *Polite*, and 9 *Impolite* examples misclassified as *Neutral*, reflecting the inherent ambiguity at the boundary between neutral and impolite speech.

5.3. Test Set Predictions

Figure 5 compares the distribution of our MARBERT model’s predicted labels on the test set against the training set label distribution. Since the ground-truth labels for the test set were not released during the competition, this comparison does not measure prediction accuracy but rather checks whether the model’s output distribution is consistent with the expected class proportions from training.

Our MARBERT model produces a prediction distribution on the test set that closely mirrors the training data proportions, suggesting well-calibrated predictions and the absence of majority-class collapse.

Confusion Matrix — MARBERT + Focal Loss (Validation Set)

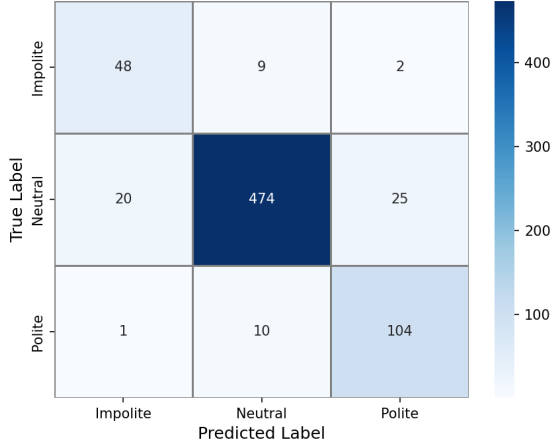


Figure 4: Confusion matrix of the best model (MARBERT + Focal Loss) on the validation set (693 samples).

This is in contrast to weaker baselines which tend to over-predict the majority *Neutral* class.

6. Discussion

Our experiments highlight several key lessons for Arabic text classification in the context of social media:

Pre-training domain matters more than model architecture. The most impactful decision in our pipeline was switching from AraBERT (news/Wikipedia) to MARBERT (tweets). Both models share a similar BERT architecture, but MARBERT’s pre-training on 1 billion Arabic tweets provides it with knowledge of dialectal expressions, informal spelling, and social media conventions that are critical for politeness detection in this domain.

Focal Loss outperforms resampling. While SMOTE (Chawla et al., 2002) is a popular approach for handling class imbalance, we find that Focal Loss applied directly during fine-tuning is more effective. This is likely because Focal Loss operates at the loss level, allowing the model to learn nuanced decision boundaries, whereas SMOTE operates at the data level and can introduce noise through synthetic interpolation in high-dimensional embedding spaces.

Simple models can outperform complex pipelines. Our best result comes from a straightforward fine-tuning approach (MARBERT + classification head + Focal Loss), without any additional architectural modifications or multi-stage pipelines. Hybrid approaches and ensembles did not improve performance, reinforcing the principle that well-matched pre-training combined with appropriate loss functions can be sufficient.

7. Conclusion

We presented our approach to the OSACT7 AdabEval Task A on Arabic politeness detection. Through systematic experimentation, we demonstrated that MARBERT fine-tuned with Focal Loss achieves strong results ($F1 = 0.84$), representing a 100% improvement over classical ML baselines. Our findings emphasize the critical importance of domain-matched pre-training for Arabic NLP tasks involving informal text. The key factors behind our system’s success are: (1) using a model pre-trained on data similar to the target domain, (2) employing Focal Loss to handle class imbalance at the loss level, and (3) careful regularization through early stopping and gradient clipping.

For future work, we plan to explore data augmentation techniques for the minority classes, investigate multi-task learning with related Arabic NLP tasks, and experiment with larger language models. We also plan to investigate the role of dialectal features in politeness classification through per-source analysis.

8. Limitations

Our work has several limitations. First, our validation results may not fully generalize to the test set, as the official test labels were not available during development. Second, the AraBERT fine-tuning results ($F1 = 0.27$) are surprisingly low — lower even than the frozen AraBERT baseline ($F1 = 0.42$). We attribute this anomaly to a combination of domain mismatch (AraBERT was pre-trained on news and Wikipedia, while our data contains informal social media text) and potential optimization instability: fine-tuning a mismatched model on a small, imbalanced dataset can lead to catastrophic forgetting or convergence to a degenerate solution. However, we cannot exclude the possibility of a hyperparameter or implementation issue, and a more thorough ablation (e.g., with learning rate sweep and class-weighted loss for AraBERT) would be needed to fully rule this out. Third, we did not perform extensive hyperparameter search due to computational constraints, and our MARBERT results could potentially be improved further with systematic tuning.

9. Bibliographical References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. [ARBERT & MARBERT: Deep bidirectional transformers for Arabic](#). In *Proceedings of the 59th Annual Meeting of the Association for*

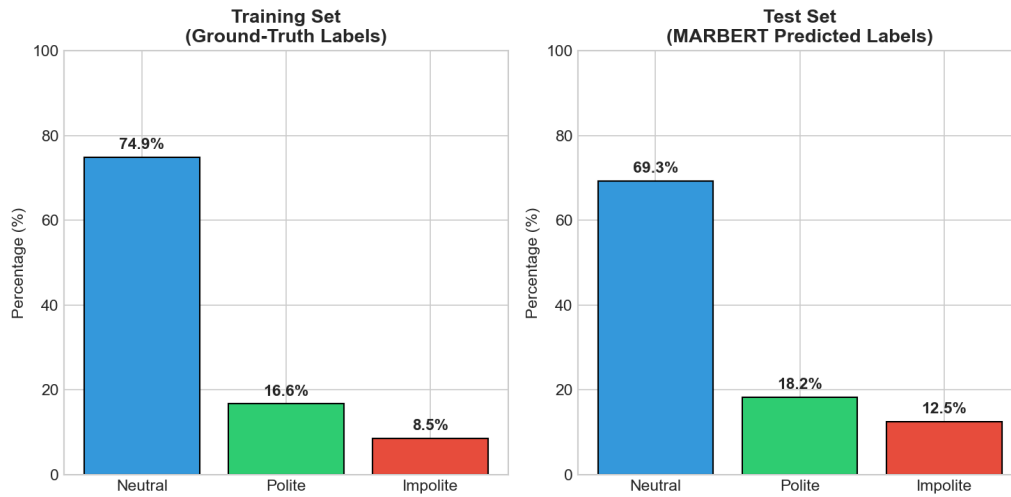


Figure 5: Comparison of class label distributions: training set (left, ground-truth labels) vs. MARBERT predictions on the test set (right, predicted labels). The similarity between the two distributions suggests that the model is well-calibrated and does not collapse to predicting the majority class. Note that ground-truth test labels were unavailable; this figure reflects the model’s predictions only.

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7088–7105. Association for Computational Linguistics.

Hend Al-Khalifa, Nadia Ghezaiel, Maria Bounnit, Hend Hamed Alhazmi, Noof Abdullah Alfear, Reem Fahad Alqifari, Ameera Masoud Almasoud, and Sharefah Al-Ghamdi. 2026. ADAB: Arabic dataset for automated politeness benchmarking — a large-scale resource for computational sociopragmatics. In *Proceedings of The 2026 International Conference on Language Resources and Evaluation (LREC2026)*, Palma, Mallorca (Spain).

Reem Alqifari, Hend Al-Khalifa, Nadia Ghezaiel Hammouda, Maria Bounnit, Hend AlHazmi, Ameera Almasoud, Sharefah AlGhamdi, and Noof Alfear. 2026. The AdabEval 2026 shared task on Arabic politeness detection. In *Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks (OSACT 2026) co-located with the 2026 International Conference on Language Resources and Evaluation (LREC2026)*, Palma, Mallorca (Spain).

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools*, pages 9–15.

Penelope Brown and Stephen C. Levinson. 1987. [Politeness: Some Universals in Language Use](#),

[age](#), volume 4 of *Studies in Interactional Sociolinguistics*. Cambridge University Press.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. [SMOTE: Synthetic minority over-sampling technique](#). *Journal of Artificial Intelligence Research*, 16:321–357.

Tianqi Chen and Carlos Guestrin. 2016. [XGBoost: A scalable tree boosting system](#). In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Cristian Danescu-Niculescu-Mizil, Moritz Suber, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. [A computational approach to politeness with application to social factors](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 250–259.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. [Focal loss for dense object detection](#). In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations (ICLR)*.

Multi-Label Classification of Arabic Politeness Criteria in Social Media

Rand Abdullah Alturki

King Saud University
Riyadh, Saudi Arabia
rand.a.t.alturki@gmail.com

Abstract

We address the problem of multi-label classification of politeness and impoliteness criteria in Arabic social media posts, as defined in Subtask B of an Arabic politeness shared task. The goal is to assign up to four labels from nine pragmatic categories, including Insult, Criticism, Respect, Prayers, and Hospitality, to each post. We first construct consistent multi-label annotations by mapping heterogeneous criterion strings into the official label set and analyzing their skewed distribution. To mitigate severe class imbalance, especially for rare categories such as Hospitality and Racism/Discrimination, we apply targeted oversampling of minority instances. Our modelling pipeline combines a TF-IDF + Logistic Regression baseline with two transformer-based encoders, MARBERT and AraBERT-twitter, trained for multi-label classification with Focal Loss. We then aggregate model outputs through a weighted ensemble and optimize per-class decision thresholds on a held-out validation set to improve macro-averaged F1. Experiments on the shared-task train/validation split show that the ensemble substantially outperforms the TF-IDF baseline and individual transformers, particularly on underrepresented categories, while maintaining competitive performance on frequent labels.

Keywords: Arabic, politeness, impoliteness, multi-label classification, social media, transformers

1. Introduction

Politeness and impoliteness are crucial aspects of interpersonal communication, and modeling them in Arabic social networks is important for applications such as content moderation, conversational agents, and sociolinguistic analysis. Subtask B of the Arabic politeness shared task (Alqifari et al., 2026) focuses on assigning up to four pragmatic criteria to each tweet, drawn from a fixed inventory of nine categories such as Insult, Criticism, Respect, Prayers, and Hospitality. Unlike standard single-label sentiment classification, this setup is inherently multi-label and exhibits strong class imbalance, with some categories occurring only a handful of times in the training data. These characteristics make the task challenging: systems must capture fine-grained pragmatic distinctions while remaining robust to sparse supervision for rare labels.

In this paper, we present a complete multi-label classification pipeline for Subtask B. We unify noisy annotation strings into the official nine categories, analyze category distributions before and after balancing, and train both traditional and neural models. Our approach combines a TF-IDF + Logistic Regression baseline with two Arabic transformer encoders, MARBERT (Ezzeldin et al., 2025) and AraBERT-twitter (Hithnawi et al., 2025), using Focal Loss (202, 2024) to better handle label imbalance. We then build a three-way ensemble and tune per-class decision thresholds to optimize macro-averaged F1 on a held-out validation set. The proposed system is simple to reproduce,

yet delivers strong performance on both frequent and rare categories.

2. Related Work

Politeness and impoliteness in computational linguistics have been studied primarily for English, often using single-label classification of requests, replies, or forum posts. Work on Arabic pragmatics has mainly focused on sentiment, toxicity, and hate speech (Alqifari et al., 2026), with relatively less attention to fine-grained politeness criteria in social media. Multi-label text classification is a well-established problem, and prior work shows that combining sparse lexical models such as TF-IDF with transformer-based encoders can yield robust performance, especially under label imbalance (Liang et al., 2025).

For Arabic social media, MARBERT (Ezzeldin et al., 2025) and AraBERT-family models (Hithnawi et al., 2025) have become strong backbones for a range of tasks, including sentiment analysis, offensive language detection, and dialect identification (Alqifari et al., 2026). Our work extends these lines by targeting multi-label politeness criteria and carefully addressing label skew through oversampling and Focal Loss (202, 2024), combined with a simple yet effective ensemble.

3. Task and Data

3.1. Task Definition

Subtask B requires assigning up to four pragmatic criteria to each Arabic tweet. The official inventory consists of nine categories: Criticism, Insult, Respect, Prayers, Greetings, Hospitality, Gratitude, Admiration / Love, and Racism / Discrimination. Each text may be associated with one to four criteria, annotated in separate columns (“criteria 1”–“criteria 4”) in the provided CSV files. The evaluation metric is macro-averaged F1 over the nine categories.

3.2. Dataset

The organizers provided us with three data splits: training, validation, and test, all with the same column structure. We focus on the train and validation sets for model development and use the test set only for final submission. The text is stored in an unnamed first column, followed by a label for politeness, and then up to four criteria columns with their associated key-word fields. For Subtask B we ignore the politeness label and key-word columns, using only the tweet text and criteria annotations.

Table 1 reports the per-category counts in the original training set, based on the unified nine-category mapping described below. The distribution is highly skewed: Respect and Insult are common, while Hospitality and Racism / Discrimination are extremely rare.

Category	Train	Val
Criticism	207	30
Insult	514	74
Respect	787	110
Prayers	415	59
Greetings	93	16
Hospitality	3	1
Gratitude	271	29
Admiration / Love	256	39
Racism / Discrimination	10	1

Table 1: Category counts in original train/validation sets (before balancing).

3.3. Label Mapping

The raw criteria columns contain a variety of strings, including bilingual labels (e.g., “Admiration & Love - الحب والإعجاب” *al-i’jāb wa al-ḥub*) and fine-grained distinctions (e.g., “Verbal violence - عنف لفظي” *‘unf lafzī*, “Threat - تهديد” *tahdīd*). We construct a mapping dictionary that collapses all observed variants into the nine official categories as shown in Table 2. For example, we map *Disparagement*,

Verbal violence - عنف لفظي (*‘unf lafzī*), *Threat* - تهديد (*tahdīd*), and *Sarcasm* - تهكم (*tahakkum*) into *Insult*, while multiple “Asking for permission” variants are mapped into *Respect*. We then parse each row by scanning the four criteria columns, applying the mapping, and collecting the resulting labels into a list.

Raw Annotation String	Mapped Category
Insult Disparagement Verbal violence (لظفي عنف, <i>‘unf lafzī</i>) Threat (تهديد, <i>tahdīd</i>) Sarcasm (تهكم, <i>tahakkum</i>)	Insult
Criticism (انتقاد, <i>intiqād</i>) Accusation	Criticism
Respect Asking for permission (الإذن طلب, <i>ṭalab al-idhn</i>) Asking for permission & affection (الإذن طلب و التودد, <i>ṭalab al-idhn wa al-tawadud</i>) Excuse (اعتذار, <i>i’tidhār</i>)	Respect
Greetings (التحية, <i>al-taḥiyya</i>)	Greetings
Hospitality & generosity (الكرم و الضيافة, <i>al-ḍiyāfa wa al-karam</i>)	Hospitality
Prayers	Prayers
Gratitude (الامتنان, <i>al-imtīnān</i>) Thanks & gratitude (الشكر و الامتنان, <i>al-shukr wa al-imtīnān</i>) Congratulatory (تهنئة, <i>tahni’a</i>) Felicitation (التهنئة, <i>al-tahni’a</i>)	Gratitude
Admiration (الإعجاب, <i>al-i’jāb</i>) Love (الحب, <i>al-ḥub</i>) Appreciation & Love (الحب و الإعجاب, <i>al-i’jāb wa al-ḥub</i>)	Admiration / Love
Racism & discrimination (تمييز و عنصرية, <i>‘unṣuriyya wa tamyīz</i>) Discrimination & sectarian (عنصرية, <i>‘unṣuriyya, taḥazzub wa tamyīz</i>)	Racism / Discrimination

Table 2: Mapping of raw annotation strings to the nine official categories. Multiple surface variants (including bilingual labels) are collapsed into a single canonical label.

Text with no valid criteria are represented by an empty label list. We use a MultiLabelBinarizer with a fixed class order to convert the label lists into 9-dimensional binary vectors for training and evaluation.

4. Methodology

4.1. System Overview

Figure 1 illustrates the overall pipeline. Raw tweet text passes through Arabic-specific preprocessing, after which three classifiers produce per-label

probability scores: MARBERT, AraBERT-twitter, and a TF-IDF + Logistic Regression baseline. Their outputs are combined via weighted averaging, and per-label decision thresholds tuned on the validation set convert the ensemble scores into final multi-label predictions.

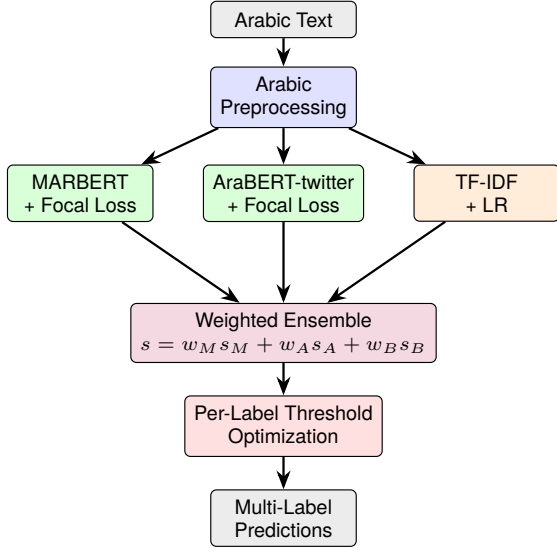


Figure 1: System pipeline for Subtask B multi-label classification.

4.2. Preprocessing

We normalize the tweet text using a light Arabic preprocessing function applied to all splits. The function replaces common character variants (e.g., mapping all forms of alef to `\ alif`, mapping `ﺀ` *alif maqsura* to `ﺀ ya'`, and normalizing hamza forms), and collapses excessive whitespace. We do not perform aggressive tokenization or stemming, relying instead on the subword tokenizers of the transformer models.

4.3. Handling Class Imbalance

The per-category counts in Table 1 show that Hospitality and Racism / Discrimination are severely under-represented, with only a few instances in the training set. To mitigate this, we perform simple oversampling at the row level. We first identify all texts whose label list includes at least one minority label (Hospitality, Racism / Discrimination, or Greetings), then replicate those rows k times and concatenate them with the original training set. We set $k = 3$, which provides a favorable balance between augmenting minority-label instances and avoiding excessive over-representation relative to the majority classes; preliminary comparisons with $k \in \{2, 5\}$ showed that $k = 3$ maximized validation macro-F1.

Figure 2 shows the category distribution before and after oversampling. The counts for Hospitality and Racism / Discrimination increase substantially, making it more feasible for the models to learn them.

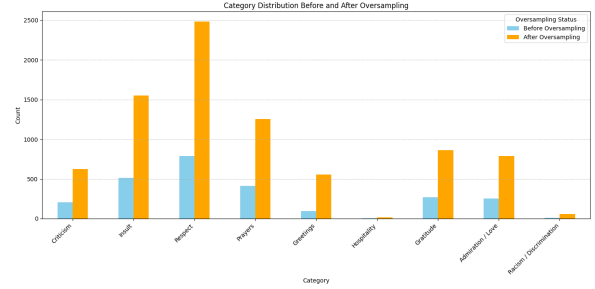


Figure 2: Category distribution before oversampling (blue) and after oversampling (orange).

4.4. Models

We experiment with three models:

- **TF-IDF + Logistic Regression:** A strong lexical baseline using a `TfidfVectorizer` with up to 20,000 features and character/word n -grams, followed by a One-vs-Rest Logistic Regression classifier.
- **MARBERT:** A transformer encoder pre-trained on large-scale Arabic social media, fine-tuned with a 9-dimensional sigmoid output layer for multi-label classification.
- **AraBERT-twitter:** A variant of AraBERT tailored to Twitter data, fine-tuned similarly with a 9-dimensional multi-label head.

For MARBERT and AraBERT-twitter, we set `problem_type="multi_label_classification"` in the Hugging Face implementation and use BCE-with-logits as the base loss.

4.5. Focal Loss

To further address label imbalance, we replace the standard BCE loss with a Focal Loss variant. Given logits z and binary targets $y \in \{0, 1\}$, we first compute the per-label BCE loss and then reweight it as:

$$\text{FL}(z, y) = \alpha(1 - p_t)^\gamma \cdot \text{BCE}(z, y),$$

where p_t is the model's probability assigned to the target class, α is a balancing factor, and γ controls the focusing strength. We adopt the standard values $\alpha = 0.25$ and $\gamma = 2.0$ which were shown to

work well across a range of imbalanced classification settings. Focal Loss down weights easy examples and emphasizes hard or misclassified examples, which is desirable in the presence of severe class imbalance.

We implement Focal Loss in a custom Trainer subclass that overrides the default loss computation and otherwise follows standard Hugging Face training.

4.6. Dataset Statistics

After label mapping and oversampling, the final training set contains 6,465 examples (vs. 2,049 original). Table 3 summarizes key statistics:

Metric	Value
Original train size	2,049
Balanced train size	6,465
Validation size	291
Test size	588
Categories	9
Avg labels per tweet	1.75
Minority class boost	3×

Table 3: Dataset statistics after preprocessing.

The validation set remains unchanged (291 examples) to ensure fair threshold tuning.

4.7. Training Setup

We fine-tune MARBERT and AraBERT-twitter for up to 8 epochs with early stopping (patience = 2) based on validation macro-F1. Key hyperparameters include:

- Batch size: 16 (training), 32 (evaluation)
- Learning rate: 2×10^{-5} with cosine decay and 10% warmup
- Weight decay: 0.01
- Max sequence length: 128 tokens
- Mixed precision: fp16 (when GPU available)

For the TF-IDF baseline, we use 20,000 features with unigram+bigram character and word n -grams, and Logistic Regression with 2,000 max iterations per binary classifier. Training takes approximately 30 minutes per model on a single NVIDIA T4 GPU (Google Colab), with MARBERT slightly faster due to its smaller vocabulary.

5. Experiments and Results

5.1. Evaluation Protocol

We treat the validation set as a held-out evaluation set. For each model, we obtain sigmoid probabilities for all nine labels and then convert them

into binary predictions using thresholds. Macro-averaged F1 is computed over the nine categories. We also report per-category F1 to highlight the effect on minority labels.

5.2. Single-Model Performance

We first evaluate the TF-IDF baseline and individual transformer models using a single global threshold per model, tuned on the validation set. Table 4 summarizes the macro-F1 scores.

Model	Macro F1
TF-IDF + Logistic Regression	0.3665
MARBERT + Focal Loss	0.2237
AraBERT-twitter + Focal Loss	0.2237

Table 4: Macro-F1 scores on the validation set for single models (global threshold).

The TF-IDF baseline provides a reasonable starting point, and at the global threshold, both transformer models underperform it. Analysis of the per-class outputs reveals that at the optimal global threshold, both MARBERT and AraBERT-twitter tend toward predicting all labels as positive (precision ≈ 0.14 , recall = 1.0 for most categories), resulting in near-zero F1 for rare classes. We attribute this to the interaction between Focal Loss and severe class imbalance: with very few positive instances per label, the models learn to predict high probabilities broadly in order to minimize focal loss. This degenerate behavior is mitigated in the ensemble through per-label threshold tuning, which allows each label to be assessed at its own calibrated operating point.

5.3. Per-Category F1 Breakdown

Table 5.3 reports per-category F1 scores on the validation set for each individual model and for the final ensemble. The ensemble consistently improves over both transformers and, for most categories, over the TF-IDF baseline as well.

Category	TF-IDF	MARBERT	AraBERT	Ensemble
Criticism	0.00	0.19	0.19	0.37
Insult	0.60	0.41	0.41	0.78
Respect	0.80	0.55	0.55	0.85
Prayers	0.71	0.34	0.34	0.85
Greetings	0.32	0.10	0.10	0.72
Hospitality	0.00	0.01	0.01	1.00
Gratitude	0.51	0.18	0.18	0.70
Admiration / Love	0.36	0.24	0.24	0.60
Racism / Discrim.	0.00	0.01	0.01	0.01
Macro Avg	0.37	0.22	0.22	0.65

Table 5: Per-category and macro-averaged F1 on the validation set. Individual transformer scores are reported at their optimal global threshold. Ensemble scores use per-label thresholds.

Per-category and macro-averaged F1 on the validation set. Individual transformer scores are reported at their optimal global threshold. Ensemble scores use per-label thresholds. TF-IDF per-category scores to be populated from final classification report.

5.4. Ensembling and Threshold Optimization

To combine the strengths of all three models, we build a linear ensemble over their probability outputs. We consider weighted combinations of the form:

$$s = w_M s_{\text{MARBERT}} + w_A s_{\text{AraBERT}} + w_B s_{\text{TFIDF}},$$

where the weights sum to 1. For each weight triple, we optimize label-specific thresholds on the validation set by searching over the grid $[0.02, 0.70]$ in steps of 0.01, selecting the threshold for each label independently by maximizing binary F1.

Table 6 reports macro-F1 for all six candidate weight triples evaluated. We select the combination that yields the best validation macro-F1.

MARBERT	AraBERT	TF-IDF	Macro F1
0.40	0.30	0.30	0.6478
0.40	0.40	0.20	0.6463
0.50	0.30	0.20	0.6473
0.30	0.30	0.40	0.6507
0.20	0.30	0.50	0.6523
0.30	0.20	0.50	0.6525

Table 6: Validation macro-F1 for all candidate ensemble weight triples. The best configuration (bold) places the highest weight on the TF-IDF baseline and lower complementary weights on the two transformers.

The final ensemble with weights $(w_M, w_A, w_B) = (0.3, 0.2, 0.5)$ achieves a macro-F1 of 0.6525 on the validation set, substantially outperforming all individual models.

5.5. Per-Category Effects

The ensemble improves F1 not only on frequent categories such as Respect and Insult, but also on minority labels. Qualitatively, MARBERT tends to provide strong representations for noisy social-media text, AraBERT-twitter complements it on some categories, and the TF-IDF baseline remains competitive for lexically distinctive labels such as Insult and Racism / Discrimination. The per-label threshold optimization reduces the tendency of the models to over-predict all labels and yields more calibrated predictions.

5.6. Test Set Evaluation

For the final submission, we apply the trained ensemble with best-found weights $(0.3, 0.2, 0.5)$ and per-label thresholds to the held-out test set. Predictions are generated following the same inference pipeline as for the validation set: text is pre-processed identically, each model produces sigmoid probabilities, scores are combined via the weighted average, and per-label thresholds derived from validation tuning are applied to produce binary predictions. Each text is assigned up to four criteria labels in descending order of ensemble score. Ground-truth labels for the test set are not available to us; final performance achieved a macro F1 of 0.41 on the provided test set.

6. Analysis and Discussion

The experiments highlight several important aspects of this task. First, we noticed that careful label mapping is essential, as many raw annotation strings are inconsistent, and collapsing them into a small, fixed category set improves both label quality and interpretability. Second, even simple oversampling of minority-label tweets significantly helps the models recognize rare categories, particularly when combined with Focal Loss. Third, the ensemble consistently outperforms individual models, showing that traditional lexical features and deep transformers capture complementary signals. In our case, the TF-IDF baseline plays a stronger role than we expected, especially for categories with highly indicative keywords.

7. Conclusion

We presented a multi-label classification system for Arabic politeness criteria in social media. Our approach combines label mapping, oversampling, transformer-based encoders with Focal Loss, and a weighted ensemble with per-label thresholds. The resulting system achieves a validation macro-F1 of 0.6525, substantially improving over individual models and particularly benefiting rare cate-

gories such as Hospitality and Racism / Discrimination. We plan to extend this work by exploring additional Arabic pre-trained models and investigating more principled calibration and threshold-tuning techniques. Future work will build on the AdabEval shared task (Alqifari et al., 2026) and ADAB dataset (Al-Khalifa et al., 2026).

Ethics Statement

This work uses publicly available Arabic social media data annotated for politeness and impoliteness criteria. Such data may contain offensive, hateful, or discriminatory content, especially in categories like Insult and Racism / Discrimination. We use the data solely for research purposes aimed at improving the detection and understanding of harmful or impolite language, and we do not attempt to deanonymize users or link posts to real identities. Models developed in this work should not be deployed without careful assessment of potential biases, including differential error rates across dialects and demographic groups.

Limitations

Our system is trained and evaluated on a relatively small dataset with a skewed label distribution, which may limit generalization to other domains or platforms. We rely on simple oversampling and Focal Loss for class imbalance, but more advanced reweighting or augmentation techniques could further improve minority-label performance. Per-class thresholds are both tuned and evaluated on the same validation set, which introduces a risk of overfitting to the validation distribution; cross-validation for threshold selection is left to future work. Finally, our analysis focuses on macro-F1 and does not fully explore other aspects such as calibration and human interpretability of the predicted criteria.

Bibliographical References

References

2024. *KSII Transactions on Internet and Information Systems*, 18(2).

Reem Alqifari, Hend Al-Khalifa, Nadia GHEZAIEL HAMMOUDA, Maria BOUNNIT, Hend AlHazmi, Ameera Almasoud, Sharefah AlGhamdi, and Noof Alfear. 2026. The adabeval 2026 shared task on arabic politeness detection. In *Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks (OSACT 2026)*

co-located with the 2026 International Conference on Language Resources and Evaluation (LREC2026), Palma, Mallorca, Spain.

Mohamed R. Ezzeldin, Gaber Sallam Salem Abdalla, and Abdoulie Faal. 2025. [Enhancing arabic sentiment analysis via <scp>marbert</scp> : Domain adaptation with pseudo-labeling and contrastive learning](#). *Engineering Reports*, 7(12).

Rania Ibrahim Hithnawi, Mohammad M N Hamarsheh, and Mohammed Maree. 2025. [Arabert for arabic cyberbullying detection in facebook comments](#). *Journal of Cybersecurity*, 11(1).

Zejian Liang, Yunxiang Zhao, Haiwen Xu, Hong Huang, and Luning Chen. 2025. [A hybrid model integrating roberta, tf-idf, and attention mechanism for medical query intent classification](#). *Scientific Reports*, 15(1).

Al-Khalifa, Hend and Ghezaiel, Nadia and Bounnit, Maria and Hamed Alhazmi, Hend and Abdullah Alfear, Noof and Fahad Alqifari, Reem and Masoud Almasoud, Ameera and Ahmed Al-Ghamdi, Sharefah. 2026. *ADAB: Arabic Dataset for Automated Politeness Benchmarking - A Large-Scale Resource for Computational Sociopragmatics*. PID <https://doi.org/10.48550/arXiv.2602.13870>.

QIAS 2026: Overview of the Shared Task on Islamic Inheritance Reasoning

**Abdessalam BOUCHEKIF¹, Somaya ELTANBOULY¹, Samer RASHWANI¹,
Shahd GABEN¹, Mutaz AL-KHATIB¹, Heba SBAHI¹,
Emad MOHAMED², Mohammed GHALY¹**

¹Hamad bin Khalifa University, Qatar ²Nazarbayev University, Kazakhstan

{abouche kif, seltanbouly, srashwani, sgaben, malkhatib, hsbahi, mghaly}@hbku.edu.qa
emad.mohamed@nu.edu.kz

Abstract

This paper presents a comprehensive overview of the QIAS 2026 shared task, organized as part of the OSACT7 Workshop and co-located with LREC 2026. The shared task was designed to evaluate the ability of large language models to perform complex reasoning in the religious and legal domain of Islamic inheritance. Unlike conventional question-answering benchmarks, QIAS 2026 focuses on end-to-end reasoning from natural language cases, requiring systems to perform the full inheritance calculation process, from identifying the eligible heirs to assigning the correct share to each beneficiary. To support this evaluation, the task was based on the MAWARITH benchmark, a dataset of 12,500 Arabic inheritance cases annotated with intermediate reasoning steps and final answers. System submissions were evaluated using MIR-E, a multi-step metric that measures performance across the main stages of inheritance reasoning. A total of 16 teams participated in the shared task, investigating a range of approaches, including prompting-based methods, retrieval-augmented generation, and fine-tuning strategies. The results show that Islamic inheritance remains a highly challenging benchmark for current language models, especially in stages that require precise legal interpretation and structured numerical reasoning. This overview summarizes the task design, dataset, evaluation framework, participating systems, and main results.

Keywords: multi-step reasoning, Islamic inheritance reasoning, Arabic language processing

1. Introduction

Large language models (LLMs) have recently achieved strong performance across a wide range of natural language processing tasks, including question answering, summarization, and complex text generation. Their success has been particularly evident in tasks that benefit from broad linguistic coverage and large-scale pretraining. However, LLMs still face important challenges in specialized domains that require precise reasoning, structured decision-making, and strict adherence to domain-specific rules. These limitations become more apparent in tasks that require a sequence of dependent reasoning steps, where an error at an early stage can propagate and compromise the final answer. This issue is especially important in religious and legal domains, where reasoning is not only knowledge-intensive but also constrained by formal principles and interpretive traditions.

In Islamic studies, and particularly in Islamic law, systems must reason over authoritative and highly structured sources such as the Qur'an, Hadith, and juristic writings. They must also operate within a framework based on clear legal principles and, in some cases, different interpretations across schools of thought. As a result, evaluating LLMs in such contexts requires benchmarks that go beyond surface-level question answering and instead test the ability of models to produce precise and

faithful reasoning. Islamic inheritance law (*ʿilm al-mawārith*) serves as an effective testbed for evaluating reasoning capabilities. This domain of Islamic jurisprudence requires multi-step legal and numerical reasoning to resolve cases. A valid solution must identify eligible heirs, determine which heirs are excluded or blocked, assign appropriate shares, assess the need for adjustments, and calculate the final distribution. The process is governed by strict jurisprudential rules, and certain cases present additional complexities such as *ʿawl* and *radd*. Due to its combination of legal interpretation, structured reasoning, and precise calculation, Islamic inheritance law offers a highly informative benchmark for assessing the reasoning abilities of modern language models. To support research in this direction, recent work introduced MAWARITH (Bouche kif et al., 2026), a large-scale benchmark of 12,500 Arabic inheritance cases designed for end-to-end reasoning from natural language descriptions. The benchmark includes detailed intermediate reasoning steps and final answers, making it possible to evaluate not only the correctness of the final output but also the validity of the reasoning process itself. This resource created an opportunity to move beyond standard multiple-choice evaluation and toward a more realistic setting in which systems must solve inheritance cases as humans would read and understand them.

The QIAS 2026 Shared Task was designed to eval-

uate whether participating systems can perform end-to-end Islamic inheritance reasoning in Arabic. It focuses on solving full cases from natural language, covering the complete reasoning process from heir identification to final share allocation. It also examines whether recent reasoning-oriented LLMs, such as Gemini, GPT, DeepSeek, Fanar, and Qwen, can transfer their strong performance on mathematical and synthetic benchmarks to the more complex domain of structured legal and religious reasoning.

In this paper, we present an overview of the QIAS 2026 Shared Task. We describe the task, the MAWARITH benchmark used in the evaluation, the MIR-E multi-step evaluation metric, the participating systems, and the main results obtained by the submitted approaches. We also discuss key lessons learned from the shared task and highlight the main challenges that current systems still face in Islamic inheritance reasoning.

2. Task Description

The QIAS 2026 Shared Task focuses on the end-to-end automation of Islamic inheritance reasoning (*‘ilm al-mawārith*) from natural language. The task requires systems to process Arabic inheritance cases and generate a complete, structured solution. For each input question, systems must produce a detailed step-by-step reasoning trace (`<think>`), followed by a concise final answer (`<answer>`). The task is formulated as a sequence of dependent reasoning stages. Systems are expected to identify and explicitly report the following components:

1. Identification of all mentioned heirs, including determining which are eligible or excluded based on applicable blocking rules (*hajb*).
2. Assignment of the correct legal shares (*furūd*) to entitled heirs according to classical Islamic inheritance jurisprudence, following the majority opinion (*al-jumhūr*).
3. Determination of whether a global adjustment to the distribution is required.
4. Computation of the final estate distribution, including cases where adjustments such as *‘awl* (proportional reduction) or *radd* (return of the residue) apply.

Participants must submit outputs in a structured format (e.g., JSON) that captures these reasoning steps. This representation enables fine-grained evaluation and allows analysis of different error types, such as legal reasoning errors versus numerical computation errors.

3. Data

The QIAS 2026 data come from the MAWARITH benchmark introduced by Boucekif et al. (2026).

It contains 12,500 cases written in Arabic and follows the majority opinion in Islamic inheritance law (*al-jumhūr*). Table 1 shows the distribution of cases across the training and test splits, as well as the complexity categories covered in the benchmark. Each case describes a complete inheritance situa-

Split	Simple	‘Awl	Radd	Total
Training	11,079	577	344	12,000
Test	456	39	5	500
Total	11,535	616	349	12,500

Table 1: Distribution of inheritance cases by legal complexity

tion in natural language. The system must identify the heirs mentioned in the case, determine which heirs are excluded by blocking rules, assign the correct legal shares, decide whether an adjustment is needed, and compute the final distribution of the estate. In this way, the benchmark does not test only final answers, but the full reasoning process required to solve an inheritance case correctly. The dataset covers a wide range of family relations found in classical Islamic inheritance law, including parents, children, spouses, siblings, grandparents, uncles, nephews, and other extended relatives. The MAWARITH dataset was built in several stages. First, inheritance cases were generated using the Almawarith calculator, which allows users to define heirs and their numbers through a structured interface and then produces the corresponding shares. This step provided a reliable foundation and helped ensure the correctness of the legal and numerical outcomes. Since the goal of MAWARITH is to evaluate reasoning from natural language, these structured cases were then rewritten as fluent Arabic inheritance questions, closer to real user queries. The outputs were then reviewed and enriched by an expert in Islamic studies, who added detailed legal and numerical explanations for each case. These explanations cover the main stages of inheritance reasoning, including heir identification, blocking rules, share assignment, and adjustment cases such as *‘awl* and *radd* when needed. To improve consistency, the expert-written explanations were standardized with the support of Gemini-2.5-Flash, while keeping the legal reasoning unchanged. Finally, the dataset was carefully validated to ensure consistency between the question, the reasoning steps, and the final inheritance shares.

3.1. MIR-E: Mawarith Inheritance Reasoning Evaluation

The QIAS 2026 shared task uses MIR-E (Mawarith Inheritance Reasoning Evaluation) (Boucekif et al., 2026), a weighted multi-stage metric for evaluating both intermediate reasoning steps and final

Team Name	Affiliations
CVPD (Swaileh et al., 2026)	CVPD Research group
Simplicity (Almansour, 2026)	King Saud University
KMS (Alkhamis, 2026)	King Saud University
QU-NLP (Alsmadi, 2026)	Qatar University
PSL (Mouhoub and Boucekif, 2026)	Paris Dauphine University
Silah (Kurdi et al., 2026)	Umm Al-Qura University
AGS-KSU (Sidaoui, 2026)	King Saud University

Table 2: Affiliations of teams that participated in the test phase and submitted a paper to QIAS 2026.

outputs in Islamic inheritance problems. Unlike standard evaluation based only on the final answer, MIR-E provides a more fine-grained assessment by scoring the main stages of the reasoning process. It includes four components:

1. **Heirs and Blocking** (S_h): evaluates whether the model correctly identifies the effective heirs, the blocked heirs, and their counts.
2. **Share Assignment** (S_s): measures whether the assigned shares for the eligible heirs are correct.
3. **Adjustment** (S_a): checks whether the model predicts the correct adjustment type (*none*, *awl*, or *radd*), and is scored only if the previous two stages are fully correct.
4. **Final Distribution** (S_f): evaluates whether the model produces the correct final distribution after the full inheritance process.

4. Results and Discussion

A total of 16 teams participated in the final phase. Table 2 summarizes the affiliations of the teams that participated in the test phase and submitted a paper. We provide a baseline implementation using Fanar-Sadiq, a modern Arabic large language model accessible via API. This baseline relies exclusively on prompting techniques, without any fine-tuning. The goal is to provide a simple yet effective reference point for evaluating model performance. The dataset and baseline code are publicly available online.¹ Overall, the submitted systems illustrate three main methodological directions: (i) end-to-end reasoning with prompting alone, (ii) fine-tuned models specialized for the task, and (iii) hybrid pipelines that combine LLM-based language understanding with deterministic symbolic reasoning. This diversity of approaches makes the shared task especially useful for comparing different strategies for Arabic legal reasoning. The participating systems explored different ways of solving Islamic inheritance cases. The most common approach explored

by participants was prompting-based end-to-end reasoning with large language models. Team PSL (Mouhoub and Boucekif, 2026) followed this setting and evaluated several models, including Gemini 2.5 Flash, Qwen3-32B, GPT-oss-120B, Llama-3.3-70B, Fanar-Sadiq, and Fanar-C-2-27B. Team KMS (Alkhamis, 2026) explored a similar direction and also evaluated Gemini 2.5 Pro and Mistral. Overall, the two teams reported similar performance trends: commercial models were generally more reliable, while open-weight models showed weaker results on this task.

Team CVPD (Swaileh et al., 2026), which achieved the best result in the shared task, proposed a RAG-based pipeline designed to generate outputs that match the MIR-E evaluation format. Their system built a knowledge base from synthetic legal question-answer pairs, with the option to include books, then retrieved relevant context for each Arabic inheritance question before generating a single JSON output. This output followed the required schema and included eligible heirs, blocked heirs, legal shares, the adjustment type when applicable, and the final *taṣīl* distribution. The system also included parsing and validation steps to ensure compatibility with the official evaluator. This approach showed that combining retrieval, controlled generation, and structured output constraints can be highly effective for this task.

Team Silah (Kurdi et al., 2026) investigated three strategies: retrieval-augmented generation based on a curated rule base, supervised fine-tuning of large language models, and a combination of fine-tuning and retrieval. Their experiments showed that fine-tuning alone performed better than retrieval-based approaches, with the best results obtained by a fine-tuned Fanar model. This suggests that task-specific fine-tuning can be an effective approach for Islamic inheritance reasoning. Team QU-NLP (Alsmadi, 2026) proposed a multi-stage Quantized Low-Rank Adaptation (QLoRA) strategy. Their approach involved initial domain adaptation on a corpus of Islamic fatwas to capture jurisprudential reasoning patterns, followed by task-specific fine-tuning on structured inheritance cases

¹https://gitlab.com/islamgpt1/qias_shared_task_2026

Rank	Team	Approach	MIR-E
1	CVPD	Hybrid RAG	0.935
2	Simplicity	Two-stage neuro-symbolic pipeline	0.931
3	KMS	Prompting-based end-to-end reasoning	0.916
4	QU-NLP	Multi-stage QLoRA fine-tuning	0.907
5	PSL	prompting-based end-to-end reasoning	0.898
6	AGS-KSU	QLoRA fine-tuning and Prompting-based end-to-end reasoning	0.84
7	Silah	LoRA fine-tuning	0.826
8	UTLM	Prompting-based end-to-end reasoning	0.742

Table 3: Official leaderboard results for QIAS 2026 for teams that participated in the test phase and submitted a paper

to optimize JSON-formatted output. This methodology allowed a relatively small 4B parameter model to achieve competitive performance, highlighting the effectiveness of specialized training strategies for complex legal reasoning tasks. Team AGS-KSU (Sidaoui, 2026) also explored a fine-tuning approach using Qwen2.5-3B, similar to the strategy adopted by Team QU-NLP. However, this fine-tuned model achieved a relatively low MIR-E score of 0.30. By comparison, their prompting-based configuration using GPT-5.4 Thinking reached 0.84, yielding a substantially stronger result.

Finally, some teams explicitly separated natural-language understanding from legal computation. Team Simplicity (Almansour, 2026) proposed a two-stage neuro-symbolic pipeline in which a commercial LLM was used only for Arabic information extraction. The extracted heirs were mapped to a standardized set of legal heir categories and then passed to a symbolic rule-based component that carried out blocking, share assignment, and final calculation according to the rules of *‘ilm al-mawārīth*. This design reflects a clear division of labor between the LLM and the symbolic module. The final performance of the submitted systems is summarized in Table 3, where Team CVPD achieved the top ranking (0.935) using a RAG pipeline based on the Qwen-9B model. Notably, the QU-NLP system’s multi-stage QLoRA approach achieved a competitive MIR-E score of 0.907. This performance is particularly significant, as it closely matches the 0.901 score of the commercial Gemini-2.5-Flash model (as reported in (Boucekif et al., 2026)). These results demonstrate that, while RAG remains a powerful tool for precision, specialized domain adaptation can enable smaller open-weight models, such as the 4B-parameter Qwen3, to achieve performance levels close to those of commercial models on this task.

5. Conclusions and Future Work

In this paper, we presented the QIAS 2026 Shared Task, designed to evaluate the ability of large lan-

guage models to solve Islamic inheritance cases end-to-end, including case understanding, heir identification, rule application, and final share distribution. The results of the participating systems revealed a clear gap between commercial and open-weight models, with commercial systems generally achieving stronger performance on this challenging reasoning task. At the same time, the findings suggest that fine-tuning LLMs on MAWARITH data can substantially improve performance, while RAG also proved beneficial for some systems.

As a future direction, we plan to organize a follow-up edition of the shared task focusing on more complex inheritance cases, including pregnancy-related cases, multiple deaths, the missing person (*mafqud*), and the intersex heir (*khunthā*). We also plan to focus on fine-tuning approaches for small Arabic LLMs. To ensure a fairer comparison across participants, future editions may restrict the use of commercial LLMs and focus on open-weight models that can be reproduced and compared under the same conditions.

6. Limitations

This work has several limitations. First, QIAS 2026 focuses only on inheritance cases based on the majority opinion (*al-jumhūr*). As a result, it does not represent the full diversity of juristic opinions in Islamic law. Second, although the benchmark covers many cases, it still does not include all the complexity of real inheritance situations. Some difficult or rare cases remain outside the current scope. Third, the MIR-E metric evaluates structured reasoning outputs in a detailed and reproducible way, but it does not fully measure explanation quality, clarity, or usefulness for real users.

Finally, some submitted systems use commercial models, which makes full reproducibility more difficult. Future work should expand the legal coverage of the benchmark and give more attention to open and reproducible systems.

7. References

- Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, et al. 2026. Fanar 2.0: Arabic generative AI stack. *arXiv preprint arXiv:2603.16397*.
- Ali Abdelaal, Mohammed Nader Al Haffar, Mahmoud Fawzi, and Walid Magdy. 2026. [Islam-icmmlu: A benchmark for evaluating llms on islamic knowledge](#). *arXiv preprint 2603.23750*.
- Nouar AIDahoul and Yasir Zaki. 2025. NYUAD at QIAS shared task: Benchmarking the legal reasoning of LLMs in arabic islamic inheritance cases. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 861–866.
- Omar Alkhamis. 2026. KMS at QIAS 2026: Evaluating llm reasoning for islamic inheritance division. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Muath Almansour. 2026. Simplicity at QIAS 2026: Decoupling language extraction from mathematical logic in islamic inheritance law. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Ameera Almasoud, Sharefah Al-Ghamdi, Reem Alqifari, Noof Alfear, and Hend Al-Khalifa. 2026. [MirathQA: A dataset for evaluating large language models on hanbali islamic inheritance reasoning tasks](#). *Data in Brief*, page 112589.
- Sanaa Alowaidi. 2025. [SEA-Team at QIAS 2025: Enhancing LLMs for question answering in islamic texts](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 940–946, Suzhou, China. Association for Computational Linguistics.
- Mohammad Alsmadi. 2026. QU-NLP at QIAS 2026: Multi-stage QLoRA fine-tuning for arabic islamic inheritance reasoning. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Salah Eddine Bekhouche, Abdellah Zakaria Selam, Hichem Telli, Cosimo Distanto, and Abdenour Hadid. 2025. [CVPD at QIAS 2025 shared task: An efficient encoder-based approach for islamic inheritance reasoning](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 929–934.
- Gagan Bhatia, Hamdy Mubarak, Mustafa Jarrar, George Mikros, Fadi Zaraket, Mahmoud Al-hirthani, Mutaz Al-Khatib, Logan Cochrane, Kareem Darwish, Rashid Yahiaoui, et al. 2026. From RAG to agentic RAG for faithful islamic question answering. *arXiv preprint arXiv:2601.07528*.
- Abdessalam Boucekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026. [MAWARITH: A dataset and benchmark for legal inheritance reasoning with LLMs](#). *arXiv preprint arXiv:2603.07539*.
- Abdessalam Boucekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouni, Aiman Erbad, and Mohammed Ghaly. 2025a. [QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860.
- Abdessalam Boucekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al-Khatib, and Mohammed Ghaly. 2025b. [Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation](#). In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 246–257.
- Iheb Chaabane, Puneesh Khanna, Suhail Mohmad, Slim Frikha, Shi Hu, Abdalgader Abubaker, Reda Alami, Mikhail Lubinets, Mohamed El Amine Seddik, Hakim Hacid, et al. 2026. Falcon-H1R: Pushing the reasoning frontiers with a hybrid model for efficient test-time scaling. *arXiv preprint arXiv:2601.02346*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *CoRR*, abs/2110.14168.
- DeepSeek AI. 2024. DeepSeek-R1: Incentivizing reasoning capability in large language models.

- Technical report, DeepSeek AI. Technical Report.
- Eman Elrefai, Mohamed Lotfy Elrefai, and Aml Hassan Esmail. 2025. [Gumball at QIAS 2025: Arabic LLM automated reasoning in islamic inheritance](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 953–959, Suzhou, China. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#).
- Shehenaz Hossain and Haithem Afli. 2025. [ADAPT–MTU HAI at QIAS2025: Dual-expert LLM fine-tuning and constrained decoding for arabic islamic inheritance reasoning](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 923–928, Suzhou, China. Association for Computational Linguistics.
- Ghader Kurdi, Hanan Justanieah, and Hala Justanieah. 2026. [Silah at QIAS 2026: Fine-tuning vs. retrieval-augmented generation for islamic inheritance reasoning](#). In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- AL-Smadi Mohammad. 2025. [QU-NLP at QIAS 2025 shared task: A two-phase llm fine-tuning and retrieval-augmented generation approach for islamic inheritance reasoning](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 892–898.
- Mohamed Motasim Hamed, Nada Ghneim, and Riad Sonbol. 2025. [HIAST at QIAS 2025: Retrieval-augmented LLMs with top-hit web evidence for arabic islamic reasoning QA](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 883–891, Suzhou, China.
- Mohamed Lamine Mouhoub and Chahinez Boucekif. 2026. [PSL at QIAS 2026: Which models perform better in arabic inheritance reasoning?](#) In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Mohamed Darwish, and Walid Magdy. 2025. [IslamicEval 2025: The first shared task of capturing LLMs hallucination in islamic content](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 480–493.
- Yossra Nouredien, Hassan Suliman, Farah Attallah, Abdelrazig Mohamed, and Sara Abdalla. 2025. [Athar at QIAS2025: LLM-based question answering systems for islamic inheritance and classical islamic knowledge](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 914–922, Suzhou, China. Association for Computational Linguistics.
- Nguyen Xuan Phuc and Thìn Đặng Văn. 2025. [PuxAI at QIAS 2025: Multi-agent retrieval-augmented generation for islamic inheritance and knowledge reasoning](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 905–913.
- Jihad R'baiti, Chouaib El Hachimi, Youssef Hmamouche, and Amal El Fallah Seghrouchni. 2025. [MorAI at QIAS 2025: Collaborative LLM via voting and retrieval-augmented generation for solving complex inheritance problems](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 947–952, Suzhou, China. Association for Computational Linguistics.
- Weizhou Shen, Ziyi Yang, Chenliang Li, Zhiyuan Lu, Miao Peng, Huashan Sun, Yingcheng Shi, Shengyi Liao, Shaopeng Lai, Bo Zhang, et al. 2025. [QwenLong-L1.5: Post-training recipe for long-context reasoning and memory management](#).
- Hicham Ghazi Sidaoui. 2026. [AGS-KSU at QIAS 2026: A comparative study of prompting and LLM approaches for structured islamic inheritance reasoning](#). In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. 2025. [OpenAI GPT-5 system card](#). *arXiv preprint arXiv:2601.03267*.
- Wassim Swaileh, Mohammed-En-Nadhir Zighem, Hichem Telli, Salah Eddine Bekhouche, Abdelah Zakaria Sellam, and Fadi Dornaika. 2026. [CVPD at QIAS 2026: RAG-guided LLM reasoning for al-mawarith share calculation and heir allocation](#). In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jesse Woo, Fateme Hashemi Chaleshtori, Ana Marasović, and Kenneth Marino. 2025. BriefMe: A legal NLP benchmark for assisting with legal briefs. *arXiv preprint arXiv:2506.06619*.

Osama Farouk Zaki. 2025. CIS-RG at QIAS 2025 shared task: Approaches for enhancing performance of LLM on islamic legal reasoning and its mathematical calculations. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 935–939.

A. Related Work

Large language models have recently been applied to a wide range of Islamic knowledge tasks, including Quranic question answering (Bhatia et al., 2026), knowledge retrieval (Mubarak et al., 2025; Phuc and Vãn, 2025), and the analysis of hallucinations in Islamic content (Mubarak et al., 2025). A recent related benchmark is *IslamicMMLU* (Abdelal et al., 2026), which evaluates LLMs on broad Islamic knowledge across the Quran, Hadith, and Fiqh through a large multiple-choice benchmark. These studies show that LLMs perform well on knowledge retrieval and basic understanding when answers rely on direct textual matching. However, they often hallucinate and show clear limitations on tasks that require structured reasoning or deep domain knowledge. In (Boucekif et al., 2025b), the authors report that several models, such as LLaMA and ALLaM, frequently cite non-existent Quranic verses or fabricate Hadith references. As a result, the generated conclusions are not only incorrect, but are also supported by false religious evidence. This behavior raises serious concerns for religious and legal applications, where correctness depends not only on the final answer but also on the authenticity and reliability of the cited sources.

To mitigate hallucinations, recent work has explored Retrieval-Augmented Generation (RAG) approaches. While RAG improves access to relevant information and enhances factual faithfulness and citation accuracy (Bhatia et al., 2026; Noureldien et al., 2025; Alowaidi, 2025), it remains insufficient for answering questions that require multi-step reasoning. This limitation has motivated the development of reasoning-oriented models that explicitly support multi-step inference. In this context, recent research has increasingly focused on such

models that aim to move beyond surface-level text generation and promote more reliable reasoning.

Models such as o3, GPT-5 (Singh et al., 2025), Gemini-2.5 (Anil et al., 2023), Gemini3, DeepSeek-R1 (DeepSeek AI, 2024), along with open models such as Fanar-C-2-27B (Abbas et al., 2026), Falcon-H1R (Chaabane et al., 2026), Fanar-Sadiq (Abbas et al., 2026) and Qwen3 (Shen et al., 2025) illustrate this trend by promoting more consistent multi-step inference through instruction tuning and large-scale pretraining. Evaluations of reasoning-oriented language models have largely focused on mathematical and logical benchmarks, on which these models have achieved strong results, particularly in arithmetic reasoning, symbolic manipulation, and competition-style mathematics (Cobbe et al., 2021; Hendrycks et al., 2021; Wei et al., 2022). Beyond mathematical and logical benchmarks, recent work has begun to investigate LLM reasoning in legally grounded settings by evaluating models on legal benchmarks such as BRIEFME (Woo et al., 2025), which require structured argumentation and rule-based reasoning. The authors show that GPT-4o can outperform human annotators on argument summarization by producing clear and coherent summaries. Even within the Islamic domain, inheritance law has received growing attention as a challenging testbed for LLM reasoning (AIDahoul and Zaki, 2025; R'baiti et al., 2025; Zaki, 2025). In particular, QIAS 2025² (Boucekif et al., 2025a) was a shared task dedicated to Islamic inheritance law (*‘ilm al-mawārith*), focusing on the evaluation of large language models under strict, rule-based legal and numerical constraints, using a benchmark of 2,200 MCQs. A similar MCQs benchmark is *MirathQA* (Almasoud et al., 2026), built from 1394 inheritance cases. Studies report that commercially deployed (e.g., Gemini and ChatGPT), reasoning-oriented models consistently outperform non-reasoning or general-purpose models on benchmarks requiring multi-step inference and structured reasoning (Boucekif et al., 2025b; Mohammad, 2025; Bekhouche et al., 2025; Motasim Hamed et al., 2025; Hossain and Afli, 2025). Additionally, Elrefai et al. (2025) show that a fine-tuned Qwen3 model achieved top-ranked performance on the QIAS 2025 shared task. However, this evaluation setup does not allow assessing whether models truly reason correctly. Models were required to select a single correct answer among six options, without any evaluation of the validity of their intermediate reasoning steps or the correctness of the legal justifications leading to that choice. Moreover, Boucekif et al. (2025b) shows that even when a model selects the correct answer, the underlying reasoning can still be incorrect or legally invalid. In contrast, the present shared task

²<https://sites.google.com/view/qias2025/>

requires models to perform end-to-end inheritance reasoning, explicitly generating intermediate reasoning steps, applying jurisprudential rules, and computing the final inheritance shares.

QU-NLP at QIAS 2026: Multi-Stage QLoRA Fine-Tuning for Arabic Islamic Inheritance Reasoning

Mohammad AL-Smadi

Qatar University
Doha, Qatar
malsmadi@qu.edu.qa

Abstract

Islamic inheritance law (علم المواريث, *ilm al-mawārith*) presents a challenging domain for evaluating large language models' structured reasoning capabilities, requiring multi-step legal analysis, rule-based blocking decisions, and precise fractional calculations. We present QU-NLP's submission to the QIAS 2026 shared task on Arabic Islamic inheritance reasoning. Our approach employs a multi-stage Quantized Low-Rank Adaptation (QLoRA) fine-tuning strategy on Qwen3-4B: (1) domain adaptation on 3,166 Islamic fatwa records to acquire inheritance terminology and jurisprudential reasoning patterns, followed by (2) task-specific training on 12,000 structured inheritance cases to optimize JSON-formatted output generation. Using 4-bit NF4 quantization with rank-128 LoRA adapters, our model achieves 90% MIR-E (Mawarith Inheritance Reasoning Evaluation) score on the test set, demonstrating competitive performance while requiring minimal computational resources. Our results show that domain-specific pre-adaptation combined with structured output training enables small language models to perform complex legal reasoning tasks effectively, matching commercial systems such as Gemini-2.5-flash.

1. Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across diverse NLP tasks (OpenAI et al., 2024), yet their ability to perform structured, rule-based reasoning under legal constraints remains insufficiently evaluated. Islamic inheritance law (علم المواريث, *ilm al-mawārith*) offers a demanding testbed: solving a case requires identifying eligible heirs, applying blocking rules, assigning Qur'anic shares, detecting adjustment mechanisms (عول, *'awl*, proportional reduction when shares exceed unity) or رد (radd, redistribution of surplus), and computing the final distribution—errors at any stage propagate deterministically.

Recent work on LLMs for Islamic knowledge tasks (Bhatia et al., 2026; Malhas et al., 2022; Mubarak et al., 2025) reveals systematic failures on structured reasoning (Boucekif et al., 2025b). Prior inheritance tasks were evaluated using MCQs dataset (Boucekif et al., 2025a; Elrefai et al., 2025; AL-Smadi, 2025; Almasoud et al., 2026), which prevent assessment of intermediate reasoning of inheritance. The QIAS 2026 shared task addresses this by requiring complete JSON reasoning traces evaluated by the multi-component MIR-E metric (Boucekif et al., 2026), enabling fine-grained error analysis impossible under answer-selection formats.

We employ multi-stage QLoRA fine-tuning on Qwen3-4B (Qwen Team, 2025), achieving 90% MIR-E—matching Gemini-2.5-flash while using a 4B parameter model, and substantially outperforming larger open-weight baselines (Boucekif et al., 2026).

2. Task and Dataset

2.1. Task Complexity

While Islamic inheritance follows deterministic jurisprudential rules, the QIAS 2026 task evaluates *end-to-end neural reasoning from natural language to structured output*. Neural models must simultaneously handle natural language understanding (parsing diverse Arabic expressions for family relationships), conditional logic through learned patterns (share assignments depending on other heirs), hierarchical blocking through pattern recognition (precedence rules such as sons blocking grandsons), conditional algorithm selection (detecting which of three adjustment types applies based on share totals and heir presence), numerical precision in text generation (fractions and percentages as exact strings), and structured output generation (syntactically valid JSON with consistent Arabic terminology).

The task difficulty arises not from the logical complexity of inheritance rules—which are well-defined—but from learning and applying them through neural pattern matching on natural language while generating structured outputs with exact numerical precision. With class imbalances (*radd* appears in only 2.8% of cases, see Section 2.2), models must generalize to unseen heir configurations without access to gold intermediate representations, making MAWARITH a demanding testbed for end-to-end legal reasoning.

2.2. MAWARITH Dataset

The MAWARITH dataset (Boucekif et al., 2026) comprises 12,500 Arabic inheritance cases follow-

ing the majority opinion (الجمهور), split into 12,000 training and 500 test instances covering 36 distinct heir categories. The majority (92.3%) are simple cases; 4.9% involve عول ('awl) and 2.8% involve رد (radd). For Stage 1 domain adaptation we additionally use 3,166 Islamic fatwa records from Islamweb¹ introducing jurisprudential terminology (الورثة *al-waratha*, الحجب *al-hajb*, العصبية *al-'asaba*) and reasoning patterns.

Split	Simple	'awl	radd	Total
Training	11,079	577	344	12,000
Test	456	39	5	500
Total	11,535	616	349	12,500

Table 1: MAWARITH dataset composition.

3. Methodology

We use Qwen3-4B (Qwen Team, 2025) fine-tuned in two stages via QLoRA (Dettmers et al., 2023) with 4-bit NF4 quantization, rank $r = 128$, $\alpha = 256$, applied to all projection layers. Stage 1 performs causal language modeling on 3,166 fatwa records (lr 210^{-4} , 2 epochs) to acquire inheritance terminology. Stage 2 trains on 12,000 structured inheritance cases (lr 310^{-5} , 6 epochs) with a system prompt enforcing JSON-only output with keys `heirs`, `blocked`, `shares`, `awl_or_radd`, `post_tasil`. Inference uses greedy decoding (temperature 0, max 1024 tokens).

Post-processing applies: (1) typographic normalization (باقي → باقي, tatweel removal); (2) structural deduplication, removing heirs appearing in both `heirs` and `blocked`; (3) label normalization, replacing unrecognised `awl_or_radd` strings with a fraction-sum inference while preserving all valid labels. A fourth *PostTasil* variant additionally recalculates `post_tasil` when identical to unadjusted `shares`.

4. MIR-E Evaluation Metric

MIR-E (Boucekif et al., 2026) decomposes inheritance reasoning into four weighted components: Heirs & Blocking (S_h), Share Assignment (S_s), Adjustment (S_a), and Final Allocation (S_f):

$$\text{MIR-E} = \alpha_h S_h + \alpha_s S_s + \alpha_a S_a + \alpha_f S_f \quad (1)$$

where $\alpha_h = \alpha_s = \alpha_f = 0.30$ and $\alpha_a = 0.10$. S_a is evaluated conditionally, receiving a non-zero score only when $S_h = S_s = 1$.

¹<https://www.islamweb.net/>

5. Results and Discussion

5.1. Overall Performance

Table 2 shows component scores on the 500-case test set. Our 4B model achieves 90.0% MIR-E, matching Gemini-2.5-flash (90.1%) while outperforming all open-weight baselines by a wide margin—including models with 8–30 more parameters.

Model	S_h	S_s	S_a	MIR-E
Gemini-2.5-flash	94.5%	92.9%	89.4%	90.1%
QU-NLP (Ours)	97.1%	94.3%	84.6%	90.0%
Qwen3-32B	69.0%	44.6%	26.5%	43.7%
GPT-OSS-120B	69.3%	32.7%	27.1%	39.1%
LLaMA-3.3-70B	64.8%	40.3%	21.5%	39.0%
Fanar-Sadiq	62.1%	36.7%	20.4%	36.8%
Fanar-C-2-27B	58.4%	31.4%	17.8%	32.8%

Table 2: Component-wise MIR-E on the 500-case test set (Basic post-processing). Baselines from (Boucekif et al., 2026).

5.2. Component-Wise and Per-Category Analysis

Heir Identification ($S_h = 97.1\%$): Highest among all systems, exceeding Gemini by 2.6pp and all open-weight baselines (58.4%–69.3%).

Share Assignment ($S_s = 94.3\%$): Exceeds Gemini by 1.4pp; multi-stage training allows Stage 1 to acquire fractional notation before Stage 2 practices assignment.

Adjustment Detection ($S_a = 84.6\%$): Trails Gemini by 4.8pp. Both adjustment types are rare ('awl 4.9%, radd 2.8%), limiting training exposure. Per-category analysis (Table 3) reveals that 'awl cases (79.2%) underperform radd cases (83.0%) despite nearly twice as many training examples (577 vs. 344 training, 39 vs. 5 test)—the bottleneck is arithmetic complexity, not data frequency. Notably, radd cases achieve 100% on both S_h and S_s , with failures confined to detection and final arithmetic, arguing directly against pattern memorization during model training.

5.3. Effect of Post-Processing

Post-processing yields a net gain of +0.2pp (89.8%→90.0%), concentrated entirely in S_h (+0.7pp) from structural deduplication—removing heirs that appeared in both `heirs` and `blocked`. Share fractions and `awl_or_radd` labels are taken directly from model output without correction. Valid labels ('awl عول, radd رد, none لا) for `awl_or_radd` are never overridden: a model may write a wrong share fraction for a residuary

Type	N	MIR-E	S_h	S_s	S_a	S_f
Simple	456	90.8%	96.4%	94.1%	86.2%	83.5%
'awl	39	79.2%	96.2%	95.1%	66.7%	50.4%
radd	5	83.0%	100%	100%	80.0%	50.0%

Table 3: Per-category MIR-E. 'Awl underperforms *radd* despite nearly twice as many training examples.

heir, causing the fraction sum to exceed 1, yet its `awl_or_radd: none`, ✗ label is still correct—correcting the label based on the sum of wrong fractions would turn a right answer into a wrong one.

The PostTasil variant produces results identical to Basic across all metrics, confirming that the model’s `post_tasil` arithmetic is correct in every case it correctly classifies the adjustment type. The 29 calculation errors (5.8%) are therefore confined to the final percentage generation step, not addressable by any post-hoc symbolic layer (see Section 5.4).

5.4. Error Analysis

To understand our model’s limitations, we conduct detailed error analysis on all 500 test cases, comparing model predictions against gold standard references across all reasoning components.

5.4.1. Error Distribution

Table 4 shows the error distribution. The three discrete failure categories (29+19+11 = 59 cases) are non-semantic: the model understands inheritance law but fails in arithmetic text generation (calculation), label precision (heir ID), or rare-event detection (*radd*). Residue label avoidance is reported separately as a systematic representational phenomenon rather than a discrete failure.

Error Type	Cases	%	Impact
Calculation	29	5.8%	−1.7pp
Residue label avoidance [†]	314	62.9%	−0.85pp
Heir Identification	19	3.8%	−0.4pp
Radd Detection	11	2.2%	−0.2pp

Table 4: Error distribution with impact on MIR-E. Discrete failures total 59 cases (11.8%); impacts are estimated. [†]Among the 417 cases requiring باقى التركة, the model substitutes an explicit fraction in 314 (75.3%). Because 83.1% of those 314 write the correct fraction, MIR-E’s tolerance absorbs most of the penalty: shares score gap = 0.045, giving $314/500 \times 0.045 \times 0.30 = -0.85\text{pp}$ (0.97 with label vs. 0.92 without). Not summed with discrete failures.

5.4.2. Error Patterns and Implications

Arithmetic vs. Semantic Errors: The most impactful errors (calculation, 29 cases, see Table 4) are *non-semantic*—the model understands inheritance law correctly but fails in final arithmetic. This suggests errors occur in the text-generation step of the final output field rather than in core reasoning, making them addressable through constrained decoding without retraining.

Complexity Degradation: Performance varies with case complexity. For simple cases involving 2–4 heirs, the model achieves approximately 91.7% MIR-E. For medium-complexity cases with 5–7 heirs, performance decreases slightly to around 88.9%, while for complex cases involving ≥ 8 mentioned heirs, it further declines to approximately 87.0%. This gradual degradation indicates a moderate impact of complexity on performance, with the model maintaining strong accuracy even in more complex scenarios.

Rare Event Underfitting: Per-category analysis (Table 3) shows that 'awl cases underperform *radd* cases (79.2% vs. 83.0%) despite having nearly twice as many more training examples. The primary bottleneck is arithmetic complexity: 'awl requires multi-fraction common-denominator computation, while *radd* requires only proportional redistribution once the type is identified. Both benefit from oversampling and explicit rule-based fallbacks as future work.

Residue Label Recall: The gold standard requires the residuary label باقى التركة in 417 of 500 evaluated cases (83.4%), reflecting the prevalence of male agnate heirs (عصبية) across the test set. The model provides this label in only 103 of those cases (24.7% recall), substituting an explicit fraction in the remaining 314 cases (75.3% avoidance rate). Table 5 summarises the breakdown.

Despite the low recall, the MIR-E cost is only −0.85pp because MIR-E’s tolerance absorbs 83.1% of avoidance cases: the model computes the numerically correct residue fraction and writes it as an explicit value—for instance, writing "7/12" when باقى التركة is expected, which falls within the evaluation tolerance. This reveals a *representational* rather than *computational* failure. Among the 314 avoidance cases, 261 (83.1%) produce the exact fraction a symbolic calculator would derive by subtracting fixed shares from unity—the model has learned to perform residue arithmetic

Residue label behaviour	Cases	Rate
Gold requires residue label	417	83.4% of test
Model provides label (recall)	103	24.7% of required
Model avoids label	314	75.3% of required
of which: correct fraction	261	83.1% of avoided
of which: wrong fraction	53	16.9% of avoided
Global MIR-E cost		−0.85pp

Table 5: Residue label recall analysis. Despite a 75.3% avoidance rate, the global MIR-E cost is only −0.85pp because 83.1% of avoidance cases compute the numerically correct fraction within the evaluation tolerance.

correctly but defaults to explicit fraction notation due to training bias toward fixed-share cases. The failure is therefore in the final token selection, not in the underlying calculation. Constrained decoding enforcing *باقى التركة* whenever an *عصبة* heir is present and fixed shares sum to less than unity would recover the correct label in the majority of affected cases without any change to the model weights.

5.5. Pipeline Success Rate

Table 6 reports cumulative success rates across the reasoning pipeline, where each row shows the percentage of cases in which all stages up to and including that point score perfectly.

Reasoning Stage	Success Rate
$S_h = 1$ (Heirs correct)	84.0%
$S_h = 1, S_s = 1$ (+ Shares correct)	81.2%
$S_h = 1, S_s = 1, S_a = 1$ (+ Adjustment)	79.4%
All stages correct	65.5%

Table 6: Cumulative pipeline success rates. Each row shows the percentage of cases where all stages up to and including that point score perfectly ($S = 1$).

While 65.5% of cases are solved perfectly across all components, the overall MIR-E reaches 90.0%. This gap is explained by the partial-credit design of MIR-E (Bouhekif et al., 2026), which assigns weighted scores to each intermediate stage ($\alpha_h = \alpha_s = \alpha_f = 0.30, \alpha_a = 0.10$). A case with correct heirs, shares, and adjustment but wrong final percentages, for instance, still earns $0.30 + 0.30 + 0.10 + 0.00 = 0.70$ MIR-E. It is worth noting that the adjustment score S_a is evaluated conditionally: it receives a non-zero value only when both $S_h = 1$ and $S_s = 1$, reflecting the sequential dependency of the reasoning pipeline. The 34.5% of imperfect cases therefore contribute meaningful partial credit, and a back-of-envelope check confirms the arithmetic: if the 65.5% perfect cases

score 1.0 and the remaining 34.5% score on average 0.71, the weighted average yields $(0.655 \times 1.0) + (0.345 \times 0.71) \approx 0.90$, consistent with our reported MIR-E.

The stage-by-stage breakdown reveals where errors enter the pipeline. The drop from heirs-correct (84.0%) to all-correct (65.5%) accumulates across three transitions: heir identification to share assignment (−2.8pp), share assignment to adjustment (−1.8pp), and adjustment to final distribution (−13.9pp). The largest single drop is the last, indicating that arithmetic computation in `post_tasil` is the dominant bottleneck for cases that pass all upstream reasoning stages—consistent with calculation errors being the most impactful error category (5.8% of cases, −1.7pp, Table 4).

6. Conclusion

We presented QU-NLP’s submission to the QIAS 2026 shared task, achieving 90.0% MIR-E through multi-stage QLoRA fine-tuning of Qwen3-4B—matching Gemini-2.5-flash (90.1%) with a 4B parameter model on consumer hardware. Error analysis identifies failures as predominantly arithmetic text-generation errors rather than reasoning deficiencies: *radd* cases reach 100% heir and share accuracy with failures confined to the final output field; PostTasil equals Basic confirming correct adjustment arithmetic in all classified cases; and 83.1% of residue-avoidance cases compute the correct fraction in explicit form—a representational failure (−0.85pp) addressable through constrained decoding without retraining. Domain adaptation on Islamic fatwa records provides jurisprudential grounding absent from general pre-training data.

Future work will focus on: (1) constrained decoding to enforce *باقى التركة* when *عصبة* heirs are present and fixed shares sum to less than unity, directly addressing the 75.3% residue avoidance rate; (2) oversampling rare adjustment cases (*radd* 2.8%, *awl* 4.9%) to reduce underfitting; and (3) extension to other Islamic legal domains and hybrid neural-symbolic architectures for safety-critical deployment.

References

- Mohammad AL-Smadi. 2025. [QU-NLP at QIAS 2025 shared task: A two-phase LLM fine-tuning and retrieval-augmented generation approach for islamic inheritance reasoning](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 892–898, Suzhou, China. Association for Computational Linguistics.
- Ameera Almasoud, Sharefah Al-Ghamdi, Reem Alqifari, Noof Alfear, and Hend Al-Khalifa. 2026. [Mirathqa: A dataset for evaluating large language models on hanbali islamic inheritance reasoning tasks](#). *Data in Brief*, 65:112589.
- Gagan Bhatia, Hamdy Mubarak, Mustafa Jarrar, George Mikros, Fadi Zaraket, Mahmoud Al-hirhani, Mutaz Al-Khatib, Logan Cochrane, Kareem Darwish, Rashid Yahiaoui, and Firoj Alam. 2026. [From RAG to agentic RAG for faithful islamic question answering](#).
- Abdessalam Boucekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026. [MAWARITH: A dataset and benchmark for legal inheritance reasoning with llms](#).
- Abdessalam Boucekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025a. [QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860, Suzhou, China. Association for Computational Linguistics.
- Abdessalam Boucekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al Khatib, and Mohammed Ghaly. 2025b. [Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation](#). In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 246–257, Suzhou, China. Association for Computational Linguistics.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLORA: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA. Curran Associates Inc.
- Eman Elrefai, Mohamed Lotfy Elrefai, and Aml Hassan Esmail. 2025. [Gumball at QIAS 2025: Arabic LLM automated reasoning in islamic inheritance](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 953–959, Suzhou, China. Association for Computational Linguistics.
- Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2022. [Qur’an QA 2022: Overview of the first shared task on question answering over the holy qur’an](#). In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur’an QA and Fine-Grained Hate Speech Detection*, pages 79–87, Marseille, France. European Language Resources Association.
- Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Mohamed Darwish, and Walid Magdy. 2025. Islamiceval 2025: The first shared task of capturing llms hallucination in islamic content. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 480–493, Suzhou, China. Association for Computational Linguistics.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, et al. 2024. [Gpt-4 technical report](#).
- Qwen Team. 2025. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.

PSL at QIAS 2026: Which Models Perform Better in Arabic Inheritance Reasoning?

Mohammed Amine Mouhoub¹ Chahinez Boucekif²

¹Paris Dauphine University

²University of Abou Bekr Belkaïd

mohamed.mouhoub@dauphine.fr

meriemchahinez.boucekif@univ-tlemcen.dz

This paper presents the participation of team PSL in the QIAS 2026 Shared Task on Arabic Islamic inheritance reasoning. The task evaluates the ability of large language models to solve inheritance cases that require legal interpretation, multi-step reasoning, and precise numerical computation. We compare *commercial* and *open-source* models under a unified prompting strategy to assess their effectiveness in structured legal reasoning with minimal task-specific adaptation.

Our results show a clear gap in reliability between the two model families. Commercial models demonstrate stronger performance in identifying eligible heirs, applying exclusion rules, and maintaining consistency across reasoning steps. In contrast, open-source models exhibit greater instability, particularly in cases involving dependent legal decisions and fractional share adjustments. The best performance is achieved by *Gemini 2.5 Flash*, with an MRE of 0.989.

1. Introduction

Large language models (LLMs) perform well on many NLP tasks, but this does not always mean that they can reason well in rule-based domains. In such tasks, a good answer must be both plausible and correct at every step. If the model makes one wrong decision early in the process, the whole solution may become incorrect. For this reason, more attention is now being given to evaluation tasks that test reasoning with structured rules, symbolic relations, and numerical accuracy.

Islamic inheritance law (*'ilm al-mawārith*) is a particularly challenging example of this problem. Solving an inheritance case requires more than retrieving legal knowledge or recognizing common patterns. A correct solution must identify the eligible heirs, apply blocking and exclusion rules correctly, assign the prescribed shares, and decide whether adjustment mechanisms such as *'awl* or *radd* are needed. Because all of these decisions are closely connected, inheritance reasoning is a useful bench-

mark for studying whether current LLMs can perform coherent multi-step legal reasoning rather than simply generate fluent legal language (Boucekif et al., 2026).

Recent work on reasoning-oriented large language models has highlighted the importance of evaluating models on tasks that require more than fluent text generation. Chain-of-thought prompting improves performance on multi-step problems by encouraging models to produce intermediate reasoning steps, while self-consistency increases robustness by selecting answers that are consistent across multiple sampled reasoning paths (Wei et al., 2022; Wang et al., 2022). More recent reasoning-focused systems, such as OpenAI's o1 family and DeepSeek-R1, further suggest that additional inference-time reasoning or reinforcement learning can lead to strong gains on difficult reasoning tasks (Jaech et al., 2024; Guo et al., 2025). However, these advances do not fully solve the problem. Reasoning remains fragile in tasks that require strict rule application, long chains of dependent decisions, or exact numerical computation, where a single intermediate mistake may invalidate the final answer (Wei et al., 2022; Wang et al., 2022).

This limitation is especially clear in specialized legal tasks such as Islamic inheritance, where correct answers depend on both legal reasoning and precise numerical calculation. In such settings, reasoning-oriented models often perform better than standard LLMs. Several studies on the QIAS 2025 benchmark (Boucekif et al., 2025a) showed that models such as Gemini and ChatGPT consistently outperformed many Arabic-focused and open-source models, including Fanar and ALLaM, on inheritance-law evaluation (Boucekif et al., 2025b; Aïdahoul and Zaki, 2025; AL-Smadi, 2025; Eddine Bekhouche et al., 2025). Notable exceptions include Elrefai et al. (2025), who obtained the best results using Qwen3, and Xuan Phuc and Đng Vãn (2025), who proposed a hybrid multi-agent architecture.

In this paper, we describe the participation of team PSL in QIAS 2026. Our submission compares commercial and open-source models under a unified prompting framework. Instead of designing a highly specialized task-specific pipeline, we examine a simpler and more informative question: how well can current general-purpose models solve a structured Arabic legal reasoning task when evaluated under the same conditions. This comparison is important because it helps clarify the gap between highly capable proprietary systems and more accessible open-weight models.

2. Data

Our experiments are conducted on the QIAS 2026 benchmark. The dataset contains 12,500 inheritance cases written in natural language and follows the majority juristic opinion (*al-jumhūr*). Each instance describes the deceased and the surviving relatives, and the task is to determine the eligible heirs and their final shares according to the rules of Islamic inheritance. Each case provides all the information required to solve the problem, including the inheriting heirs, blocked heirs, assigned shares, the possible application of *'awl* or *radd*, and the final normalized distribution. In addition, the dataset includes an intermediate reasoning trace and a concise final answer, making it suitable for evaluating both reasoning quality and final prediction accuracy. The full corpus is divided into 12,000 training instances and 500 test instances. It covers 36 distinct heir categories, ranging from close relatives such as parents, children, and spouses to more distant agnatic relatives across multiple generations. The cases vary in difficulty, from simple configurations with a small number of heir types to more complex scenarios involving up to twelve distinct heir categories. In terms of legal complexity, the training split contains 11,079 simple cases, 577 *'awl* cases, and 344 *radd* cases. The totals reported in the paper imply that the 500-case test split contains 456 simple cases, 39 *'awl* cases, and 5 *radd* cases. This distribution shows that most examples do not require adjustment, while a smaller but important subset evaluates the model's ability to handle more difficult redistribution and proportional-reduction cases.

3. System Overview

Our submission uses a simple prompting-based chain-of-thought (CoT) design. The goal is not to build a complex task-specific system, but to compare different types of models in a fair setting. In particular, we compare reasoning and non-reasoning models, as well as commercial and non-commercial models, under the same conditions. This allows us to better assess their actual ability

to solve Arabic inheritance reasoning problems. Given an inheritance case in Arabic, the system prompts the model to produce a structured solution. The prompt encourages the model to proceed through the main inheritance stages: identifying heirs, applying blocking rules, determining legal shares, and producing the final answer. To reduce unnecessary variation in generation, we use a constrained output format and a small amount of post-processing.

3.1. Models

We evaluated both commercial and open-source models to better understand the current gap between these two groups on a structured Arabic legal reasoning task. The models were selected for three main reasons. First, we wanted a fair comparison between systems that are widely accessible to researchers. Second, we wanted diversity in model families and capabilities. Third, we wanted to test whether strong general reasoning models can transfer well to a domain that combines Arabic input, legal constraints, and precise numerical output.

3.1.1. Commercial Models

Our commercial systems included *Gemini 2.5 Flash* and *Gemini 2.5 Pro*. These models were selected for their strong general-purpose reasoning capabilities and solid performance in applied NLP tasks. They provide a realistic baseline for systems that users may rely on when addressing legal or knowledge-intensive problems without task-specific fine-tuning.

3.1.2. Open-Source Models

Our open-source systems included *Qwen3-32B*¹, *GPT-oss-120B*², *Llama-3.3-70B*³, *Fanar-Sadiq* and *Fanar-C-2-27B*⁴. These models were chosen to cover different profiles, such as multilingual instruction models, Arabic-oriented systems, and large open-weight reasoning models. We also conducted preliminary experiments with *Falcon*⁵; however, its outputs frequently failed to follow the prompt instructions, resulting in very poor performance. Consequently, it was excluded from the final evaluation.

¹<https://huggingface.co/Qwen/Qwen3-32B>

²<https://huggingface.co/openai/gpt-oss-120b>

³<https://huggingface.co/meta-llama/Meta-Llama-3-70B>

⁴<https://api.fanar.qa/docsdescription/introduction>

⁵<https://huggingface.co/tiiuae/Falcon3-10B-Instruct>

Model	Type	Score
Gemini-2.5-Pro	Commercial	0.931
Gemini-2.5-Flash 2*	Commercial	0.898
Qwen3-32B	Open-source	45.1
GPT-OSS-120B	Open-source	38.7
LLaMA-3.3-70B	Open-source	35.9
Fanar-Sadiq	Open-source	35.8
Fanar-C-2-27B	Open-source	33.1

Table 1: Main results for the PSL systems on QIAS 2026. * indicates the official submission.

3.2. Prompting Strategy

Our method is based on a unified prompting framework. Each model receives the inheritance case in Arabic together with instructions asking it to solve the case in a structured way. The prompt explicitly encourages the model to reason step by step and to return an answer that follows a predefined format.

In practice, the prompt asks the model to:

1. identify the heirs who inherit,
2. apply relevant exclusion or blocking rules,
3. assign the correct legal shares,
4. and produce the final distribution.

We intentionally kept the prompt relatively simple. The goal was not to encode a handcrafted expert system inside the prompt, but rather to observe how much structured reasoning the models can perform under shared instructions.

For all models, we required the output to show the reasoning at each step, followed by a final answer in a structured format that matches the evaluation metric. This made the outputs more consistent, easier to process, and helped us see exactly where any reasoning mistakes occurred. We only applied simple post-processing to ensure that each heir was represented in a single, uniform form. We did not use extensive correction rules, since our goal was to evaluate the models’ abilities rather than to engineer a complex pipeline.

4. Results and Analysis

Table 1 reports the overall MIR-E scores obtained by all evaluated models. Overall, the results reveal a substantial performance gap between commercial and open-source models (Table 1). Commercial models achieve significantly higher scores, with *Gemini-2.5-Pro* reaching 0.931 and *Gemini-2.5-Flash 2* (official submission) achieving 0.898. In contrast, open-source models perform considerably worse, with scores ranging from 33.1 to 45.1. The strongest open-source model, *Qwen3-32B*,

achieves 45.1, remaining more than 50 points below the official system. Furthermore, performance differences among open-source models are relatively limited, suggesting a broadly similar level of capability within this group.

Beyond aggregate performance, the comparison highlights important qualitative differences in reasoning behavior. Commercial models demonstrate a stronger ability to maintain coherent multi-step reasoning, preserving consistency across heir identification, exclusion rules, and final share allocation. They are also more robust when early-stage decisions are ambiguous or complex. In contrast, open-source models frequently exhibit structural errors at early stages of the reasoning process. Such errors tend to propagate, leading to incorrect intermediate steps and ultimately invalid final distributions. For example, the omission of a valid heir often results in incorrect redistribution of shares.

A further observation is that surface-level fluency does not reliably indicate correctness. Several models produce outputs that appear legally plausible, yet contain latent structural inconsistencies, such as incorrect application of exclusion rules or infeasible share allocations.

Taken together, these results suggest that commercial models are both more accurate and more stable in handling structured legal reasoning tasks, whereas open-source models remain limited in their ability to sustain coherent reasoning across interdependent decision steps. For comparison, we also consider our method as a participating system in the shared task.

For comparison, we also report the scores of the systems that participated in the shared task. Table 2 presents the official ranking of the participating teams. The proposed methods cover a wide range of approaches, from simple prompting-based techniques to fine-tuning strategies that adapt language models to the inheritance task. Some approaches rely on Retrieval-Augmented Generation (RAG) to find similar cases and support reasoning, while others adopt hybrid methods that combine large language models (LLMs) with rule-based systems.

4.1. Common Error Types

Our manual inspection identified several recurring error patterns:

Missing or hallucinated heirs. Some models failed to identify all heirs mentioned in the scenario, while others introduced relatives who were not present in the case. This problem is especially harmful because all later steps depend on the correct heir set.

Rank	Team	MIR-E
1	CVPD (Swaileh et al., 2026)	0.935
2	Simplicity (Almansour, 2026)	0.931
3	KMS (Alkhamis, 2026)	0.916
4	QU-NLP (Alsmadi, 2026)	0.907
5	PSL (our submission)	0.898
6	AGS-KSU (Sidaoui, 2026)	0.84
7	Silah (Kurdi et al., 2026)	0.826
8	UTLM	0.742

Table 2: Official leaderboard results for QIAS 2026 for teams that participated in the test phase and submitted a paper

Incorrect blocking decisions. Another common issue was the wrong application of exclusion rules. In some outputs, an heir who should have been blocked was retained; in others, a valid heir was incorrectly removed.

Share assignment errors. Even when the correct heirs were identified, some models assigned the wrong legal shares. These errors often reflected confusion between similar family configurations or incomplete understanding of conditional inheritance rules.

Arithmetic inconsistency. Some outputs contained legal-sounding reasoning but inconsistent calculations. In such cases, the fractions did not sum correctly, the normalization was missing, or the final per-heir allocation contradicted the earlier explanation.

Format instability. A smaller but still relevant issue was output instability. Some models ignored the requested format, mixed reasoning and answer sections unpredictably, or returned partial solutions.

This also explains why good instruction-following is not enough. A model can follow the prompt and produce a clear and convincing explanation, but still make mistakes in the legal structure of the solution.

Another possible reason for the low performance of Arabic-oriented models is that they are not trained on this type of structured reasoning. Inheritance problems require specific logical steps and rule-based decisions, which may not be well represented in their training data. In this context, the QIAS 2026 dataset can be very useful. It provides examples of Islamic inheritance reasoning that could help models learn this type of task and improve their performance.

5. Conclusion

This paper presented the participation of team PSL in the QIAS 2026 Shared Task on Arabic Islamic inheritance reasoning. Our submission focused on a controlled comparison between commercial and open-source language models under a shared prompting framework.

The experiments showed that commercial models remain clearly stronger on this task, especially when the case requires several dependent legal decisions and precise numerical consistency. Open-source models show promise, but they still suffer from frequent structural errors that propagate across the reasoning chain. More broadly, our findings confirm that Islamic inheritance is a demanding testbed for evaluating legal reasoning in Arabic.

In future work, we plan to explore stronger output constraints, step-level verification, and domain-adapted training strategies to improve consistency on this class of tasks. We also believe that combining LLMs with explicit reasoning checks may be a promising direction for legal and rule-based NLP more generally.

Acknowledgments

We would like to thank the organizers of QIAS 2026 for their efforts in designing and running the shared task. In particular, we are grateful to Dr. Abdessalam Boucheikif for his guidance and support.

Mohammad AL-Smadi. 2025. [QU-NLP at QIAS 2025 shared task: A two-phase LLM fine-tuning and retrieval-augmented generation approach for islamic inheritance reasoning](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 892–898, Suzhou, China. Association for Computational Linguistics.

Nouar AIDahoul and Yasir Zaki. 2025. [NYUAD at QIAS shared task: Benchmarking the legal reasoning of LLMs in Arabic islamic inheritance cases](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 861–866, Suzhou, China. Association for Computational Linguistics.

Omar Alkhamis. 2026. KMS at QIAS 2026: Comparing LLM reasoning and hybrid symbolic solving for islamic inheritance division. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.

- Muath Almansour. 2026. Simplicity at QIAS 2026: Decoupling language extraction from mathematical logic in islamic inheritance law. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Mohammad Alsmadi. 2026. QU-NLP at QIAS 2026: Multi-stage QLoRA fine-tuning for arabic islamic inheritance reasoning. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Abdessalam Boucekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026. [Mawarith: A dataset and benchmark for legal inheritance reasoning with llms](#). *arXiv preprint arXiv:2603.07539*.
- Abdessalam Boucekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025a. [QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860.
- Abdessalam Boucekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al Khatib, and Mohammed Ghaly. 2025b. [Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation](#). In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 246–257.
- Salah Eddine Bekhouche, Abdellah Zakaria Sellam, Telli Hichem, Cosimo Distanti, and Abdenour Hadid. 2025. [CVPD at QIAS 2025 shared task: An efficient encoder-based approach for islamic inheritance reasoning](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 929–934, Suzhou, China. Association for Computational Linguistics.
- Eman Elrefai, Mohamed Lotfy Elrefai, and Aml Hassan Esmail. 2025. [Gumball at QIAS 2025: Arabic LLM automated reasoning in islamic inheritance](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 953–959, Suzhou, China. Association for Computational Linguistics.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Ghader Kurdi, Hanan Justanieah, and Hala Justanieah. 2026. Silah at QIAS 2026: Fine-tuning vs. retrieval-augmented generation for islamic inheritance reasoning. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Hicham Ghazi Sidaoui. 2026. AGS-KSU at QIAS 2026: A comparative study of prompting and LLM approaches for structured islamic inheritance reasoning. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Wassim Swaileh, Mohammed-En-Nadhir Zighem, Hichem Telli, Salah Eddine Bekhouche, Abdellah Zakaria Sellam, and Fadi Dornaika. 2026. CVPD at QIAS 2026: RAG-guided LLM reasoning for al-mawarith share calculation and heir allocation. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with LREC-COLING 2026*, Palma de Mallorca, Spain.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Nguyen Xuan Phuc and Thìn Đng Văn. 2025. [PuxAI at QIAS 2025: Multi-agent retrieval-augmented generation for islamic inheritance and knowledge reasoning](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 905–913, Suzhou, China. Association for Computational Linguistics.

AGS-KSU at QIAS 2026: A Comparative Study of Prompting and LLM Approaches for Structured Islamic Inheritance Reasoning

Hicham Ghazi Sidaoui

King Saud University (AGS-KSU)

hicham.gs@hotmail.com

Abstract

This paper describes our submission to the QIAS 2026 shared task on structured Islamic inheritance reasoning, based on the MAWARITH benchmark (Boucekif et al., 2026). The task requires multi-step structured prediction for Arabic inheritance cases, including heir identification, blocking, share assignment, adjustment detection, and final distribution, evaluated with the MIR-E metric. We compare four system configurations: a QLoRA fine-tuned Qwen2.5-3B baseline, a multi-stage Fanar-Sadiq pipeline with deterministic validation and post-processing, and two GPT-5.4 prompting setups. On the official test set, the best result was achieved by GPT-5.4 with explicit inheritance rules and development examples used as in-context demonstrations, reaching a MIR-E score of 0.84, compared with 0.76 for a minimal-prompt GPT-5.4 variant. These results suggest that explicit rule conditioning and in-context demonstrations can improve performance in this setup. Since the compared systems vary in model family and prompting strategy, the findings should be interpreted as a comparison of task configurations rather than a controlled model-only comparison.

Keywords: Islamic inheritance law, Arabic NLP, structured output generation, prompting, in-context learning, legal reasoning, QLoRA

1. Introduction

Islamic inheritance law (*Ilm al-Faraid*) requires precise multi-step reasoning: identifying eligible heirs, determining which relatives are blocked, assigning Quranic fractional shares, detecting proportional adjustment, and computing the final distribution. Because each stage depends on the correctness of preceding ones, the task is a challenging structured reasoning benchmark for NLP systems.

The QIAS 2026 shared task is based on the MAWARITH benchmark (Boucekif et al., 2026), requiring end-to-end structured output generation across the main inheritance reasoning stages, evaluated with the MIR-E metric. Unlike the earlier QIAS 2025 task (Boucekif et al., 2025a), which was framed as multiple-choice QA, QIAS 2026 demands full structured prediction. Prior work showed a substantial performance gap between large proprietary models and Arabic-specific models (Boucekif et al., 2025b), and that retrieval-augmented fine-tuning could yield strong results (AL-Smadi, 2025). These findings motivate our investigation of fine-tuning and prompting-based approaches.

This paper evaluates four engineered system configurations that trade off supervision type, cost, controllability, and inference-time structure enforcement: (1) a QLoRA fine-tuned Qwen2.5-3B baseline, (2) a multi-stage Fanar-Sadiq inference pipeline, (3) a GPT-5.4 configuration with a lightweight inference template, and (4) a GPT-5.4 configuration with explicit legal rules and in-context demonstrations.

2. Related Work

The MAWARITH benchmark (Boucekif et al., 2026) introduces the MIR-E metric, which scores each reasoning stage independently with weighted aggregation. The QIAS 2025 shared task (Boucekif et al., 2025a) attracted dozens of teams; Boucekif et al. (2025b) show that performance gaps reflect reasoning ability rather than Arabic language proficiency alone. System papers from QIAS 2025 include Bekhouche et al. (2025), who used encoder-based relevance scoring; AL-Smadi (2025), who combined LoRA fine-tuning with RAG; and Aldahoul and Zaki (2025), who showed ensemble voting outperforms individual models. QLoRA (Detrmers et al., 2023) and in-context learning (Brown et al., 2020) are the core technical methods we employ.

3. Task and Evaluation

The MAWARITH dataset (Boucekif et al., 2026) contains 12,500 Arabic inheritance cases. The shared-task split provides training data, 200 development examples with gold annotations, and 500 test examples. Outputs are structured JSON covering eligible heirs, blocked heirs, legal shares, adjustment status (*awl/radd*), and final distribution. Systems are scored with MIR-E, using set-based F1 for heir stages and value-based matching for share and distribution stages. All systems used only resources released by the shared-task organizers; no external labeled inheritance data or manual correction of test predictions was used.

4. System Configurations

4.1. Qwen2.5-3B with QLoRA

We fine-tuned Qwen2.5-3B-Instruct (Qwen Team, 2024) using QLoRA (Dettmers et al., 2023) on a 2,000-example subset of the training split sampled with fixed random seed 42, due to time and hardware constraints on an NVIDIA RTX 4060 Laptop GPU (8GB VRAM). Training used 4-bit NF4 quantization with bfloat16 compute. The LoRA adapter used rank $r = 16$, $\alpha = 32$, dropout 0.05, applied to all seven attention and MLP projection layers. Training ran for 3 epochs, sequence length 1,024, batch size 1 with gradient accumulation 4 (effective 4), learning rate 2×10^{-4} with cosine schedule and 20 warmup steps, paged AdamW 8-bit optimizer, and gradient checkpointing. Training completed in approximately 5.5 hours, reaching a final training loss of 0.26.

At inference, the model tended to generate extended Arabic reasoning before producing JSON, exhausting the generation budget. Prefilling the assistant turn with an opening brace and disabling the thinking template reduced generation time from ~ 170 s to ~ 25 s per example. Despite this, structured-generation failures resulted in a MIR-E score of 0.30. We include this configuration as a resource-constrained fine-tuning baseline rather than a performance-optimized local model.

4.2. Fanar-Sadiq Pipeline

Fanar-Sadiq (Abbas et al., 2025) is an Arabic-focused LLM accessed via API. We implemented a three-stage inference pipeline: (1) heir extraction, (2) fiqh reasoning for share assignment and adjustment, and (3) constrained structured JSON extraction. All three stages used the same Fanar backend with temperature 0.0.

The pipeline processed all 500 test examples sequentially, with each example requiring three API calls. The heir-extraction prompt asked the model to identify all relatives mentioned in the case text and normalize their names to canonical forms. The reasoning prompt then took the extracted heir list and requested full fiqh reasoning covering share assignment, blocking decisions, and adjustment detection. Finally, the structured-extraction prompt took the reasoning output and asked for a JSON response conforming to the six-field task schema, with no legal reasoning included in the output.

After structured extraction, we applied a deterministic validation-and-repair layer, including heir-name normalization, completeness enforcement for the heirs field, recomputation of `post_tasil` percentages, and deterministic computation of `awl_stage` when needed. If the extracted JSON

failed validation, the structured-extraction stage was retried once using validation errors as corrective feedback. At the batch-execution level, API requests used exponential-backoff retry handling for rate-limit failures. Fanar-Sadiq achieved MIR-E 0.46.

4.3. GPT-5.4

We evaluated two inference configurations of GPT-5.4, a commercial large language model accessed via API, under otherwise identical decoding settings: temperature 0, top_p 1.0, frequency and presence penalty 0.0, with batches of five test cases per request.

The **minimal-prompt** variant used a brief instruction to solve each inheritance case and return the answer in the required structured JSON format, including heirs, blocked heirs, shares, `awl_or_radd`, optional `awl_stage`, and `post_tasil`. It did not include explicit legal rules or in-context examples. This configuration achieved MIR-E 0.76.

The **full-prompt** variant used the same target JSON schema, but added two components: (1) an explicit rule block covering eight hard-case families (grandfather-sibling share selection, sister-with-daughters agnatic transformation, *radd* exclusion of spouses, multiple grandmother blocking, *al-Akdariyyah*, *al-Mushtarakah*, son's-daughter interactions, and the *Umariyyatan* rule), and (2) 20 gold development examples per batch as fully solved JSON demonstrations, selected by fixed circular offset: batch i used examples $[(20i) \bmod 200, \dots, (20i + 19) \bmod 200]$. The exact prompt is embedded in the code in the repository. The development examples were used only as inference-time in-context demonstrations and were not used for parameter updates. We use "in-context learning" to refer to inference-time prompting with fully solved development examples, without any parameter updates. Following standard shared-task practice, the released development set was used only for inference-time prompting and not for parameter updates or test-set selection. This configuration achieved MIR-E 0.84.

Post-processing was deterministic and restricted to schema-level normalization, including count-annotation removal from heir names, required-field completion, and fraction-format verification. No legal predictions were modified.

5. Results and Analysis

Table 1 summarizes the official test-set results. The GPT-5.4 full-prompt configuration achieved the highest MIR-E of 0.84. Because the two GPT-5.4 configurations differed only in inference tem-

System	Method	MIR-E
Qwen2.5-3B (fine-tuned)	QLoRA	0.30
Fanar-Sadiq pipeline	Zero-shot	0.46
GPT-5.4 + minimal prompt	Prompting	0.76
GPT-5.4 + rules + dev examples	In-context	0.84

Table 1: Official test-set MIR-E for the four submitted system configurations.

plate content, the improvement from 0.76 to 0.84 provides a more controlled within-model comparison of prompt design, since model family and decoding settings were held fixed across the two GPT-5.4 runs. In this narrower sense, the observed improvement is consistent with a positive contribution from explicit rule conditioning and in-context demonstrations in our setup. These results should be interpreted as a comparison of submitted system configurations rather than a controlled cross-model study, because prompting regimes and supervision levels were not matched across model families.

6. Error Analysis

To better understand the remaining gap between high MIR-E performance and exact structured correctness, we compared the 0.84 configuration’s predictions against the released gold file across four components (Table 2). The system is strongest on *awl_or_radd* (93.6%) and weaker on exact heir and distribution matching. Table 3 summarizes the main error categories.

Component	Matches	Rate
Heirs list	350/500	70.0%
Blocked list	345/500	69.0%
<i>awl_or_radd</i>	468/500	93.6%
Normalized final dist.	355/500	71.0%
All four jointly	284/500	56.8%

Table 2: Diagnostic exact-match statistics (more stringent than MIR-E; diagnostic only).

Error category	Cases	Rate
Grandmother blocking errors	63	12.6%
False <i>radd</i> (residual agnate removed)	37	7.4%
<i>awl_or_radd</i> mismatches	32	6.4%
Collateral continuation errors	20	4.0%

Table 3: Top error categories out of 500 test cases.

Error categories were assigned by automated comparison of predicted outputs against gold annotations, with each case labelled according to

Case	Gold	Prediction	Error
nh3a2q4h_4	Fixed shares + paternal uncles (residual)	Uncles blocked; <i>radd</i> applied	False <i>radd</i>
nh2s1u5s_7	Mother, paternal half-sister, and residual collateral heir	Residual collateral heir blocked	Collateral error

Table 4: Representative prediction errors made by the 0.84 system on the test set.

its primary observed failure type. The dominant errors include **grandmother-line confusion** (63 cases), where grandmother eligibility or blocking differs between the gold and predicted outputs; **false *radd*** cases, where the prediction redistributes the remainder despite the gold output preserving a residual agnate (e.g., case nh3a2q4h_4); ***awl_or_radd* mismatches** (32 cases), mostly involving *radd* with non-spousal heirs; and **collateral continuation errors**, where the gold output preserves a collateral agnatic heir but the prediction blocks that heir (e.g., case nh2s1u5s_7).

These comparisons suggest that the system’s remaining weaknesses are concentrated in legally difficult remainder-allocation decisions. In many cases, the prediction captures part of the heir structure correctly, but fails in the final legal resolution step: deciding whether a residual agnate remains, whether *radd* is permitted, and whether the remainder should continue to a collateral heir. Future improvements should focus on explicit rule enforcement for grandmother precedence, residual agnate detection, and remainder transfer.

7. Conclusion

This paper described our submitted system configurations for the QIAS 2026 shared task. Among them, GPT-5.4 with explicit inheritance rules and in-context demonstrations achieved the highest MIR-E of 0.84. More broadly, structured Islamic inheritance reasoning appears to be a demanding benchmark for Arabic legal AI, as it requires compositional rule application, exact arithmetic, and multi-stage output consistency simultaneously. The results suggest that rule-aware prompting, in-context demonstrations, and schema-aligned inference design were plausible contributing factors in the strongest evaluated configuration. Because the evaluated systems were not matched for supervision level or prompting regime, the results should be interpreted at the configuration level rather than as evidence about

model family in isolation. Future work should investigate matched cross-model comparisons, targeted handling of hard cases (grandmother blocking, residual agnates, *radd* interactions), and generalization to other *madhahib*.

8. Ethics Statement

This work involves Islamic inheritance law, a sensitive religious and legal domain. The system is intended for research purposes only and should not be used as a substitute for qualified legal or religious advice. All data were provided by the QIAS 2026 shared-task organizers.

9. Data and Code Availability

All materials are publicly available at <https://github.com/xAGS1/qias2026-ags-ksu>, including prompt templates, inference scripts, post-processing code, and final prediction files. The prediction file can be independently scored using the official MIR-E evaluator to verify the 0.84 result.

10. Acknowledgements

The author thanks the QIAS 2026 shared task organizers for providing the MAWARITH benchmark and evaluation infrastructure.

11. Bibliographical References

Hanan Abbas et al. 2025. Fanar: A large language model for arabic. Technical report, Qatar Computing Research Institute.

Mohammad AL-Smadi. 2025. QU-NLP at QIAS 2025 shared task: A two-phase LLM fine-tuning and retrieval-augmented generation approach for islamic inheritance reasoning. *arXiv preprint arXiv:2508.15854*.

Nouar AlDahoul and Yasir Zaki. 2025. Benchmarking the legal reasoning of LLMs in arabic islamic inheritance cases. *arXiv preprint arXiv:2508.15796*.

Salah Eddine Bekhouche et al. 2025. CVPD at QIAS 2025 shared task: An efficient encoder-based approach for islamic inheritance reasoning. *arXiv preprint arXiv:2509.00457*.

Abdessalam Boucekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026. MAWARITH: A dataset and

benchmark for legal inheritance reasoning with LLMs. *arXiv preprint arXiv:2603.07539*.

Abdessalam Boucekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghoulani, Aiman Erbad, and Mohammed Ghaly. 2025a. *QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment*. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860.

Abdessalam Boucekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al Khatib, and Mohammed Ghaly. 2025b. *Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation*. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 246–257.

Tom B. Brown et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Silah at QIAS 2026: Fine-Tuning vs. Retrieval-Augmented Generation for Islamic Inheritance Reasoning

Ghader Kurdi, Hanan Justanieah, and Hala Justanieah

Department of Data Science, College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia

Department of Jurisprudence, College of Shari'ah and Islamic Studies Umm Al-Qura University,
Makkah, Saudi Arabia

The Higher Division, Institute of the Holy Grand Mosque in Makkah, Makkah, Saudi Arabia

{grkurdi, hmjustaniah}@uqu.edu.sa, dr.hala.just@gmail.com

Abstract

Islamic inheritance is a highly structured, rule-intensive domain that requires precise reasoning. The QIAS 2026 shared task introduces a benchmark for evaluating generative artificial intelligence on end-to-end inheritance problem solving. In this paper, we present our team Silah's participation, comparing three approaches: 1) a multi-stage retrieval-augmented, rule-guided pipeline, 2) supervised fine-tuning of generative large language models, and 3) retrieval-augmented fine-tuning. We evaluate several open-source models, including Qwen2.5, LLaMA, DeepSeek, and Fanar. Our results show that supervised fine-tuning consistently outperforms retrieval-based approaches, with the fine-tuned Fanar-1-9B-Instruct model achieving the best performance (MIR-E = 0.83). These findings suggest that learning implicit reasoning patterns through fine-tuning is more effective than explicit rule injection under current retrieval setups, highlighting the need for more precise and selective rule retrieval in future approaches.

Keywords: Islamic Inheritance, Large Language Models, Generative AI, RAG, Fine-Tuning

1. Introduction

The Islamic law of inheritance, known as Ilm al-Fara'id or al-Mawarith, is a fundamental branch of Islamic jurisprudence. It is characterised by a highly structured and requires precise reasoning to determine heirs, assess eligibility, and allocate shares. In addition, the distribution of shares may involve mathematical adjustments, increasing complexity. As such, it provides a well-defined yet challenging domain for evaluating advanced reasoning capabilities in artificial intelligence (AI).

The Question-and-Answer in Islamic Studies Assessment (QIAS 2026) shared task provides a benchmark for evaluating generative AI models in Islamic inheritance problem solving. The task requires interpreting natural-language scenarios and producing structured outputs that include identified heirs, blocked heirs, assigned shares, adjustment mechanisms such as awl and radd, and the final distribution of the estate. This highlights the dual challenge of linguistic understanding and adherence to domain-specific rules.

In this work, team Silah (صلة) explores three generative AI approaches for the QIAS 2026 shared task: (1) a multi-stage retrieval-augmented, rule-guided pipeline (RAG-P) based on retrieval-augmented generation (RAG) (Lewis et al., 2020), where relevant rules are dynamically retrieved and injected into the prompt; (2) supervised fine-tuning (SFT) on the QIAS dataset; and (3) retrieval-augmented fine-tuning (RAG-FT), which follows the same training procedure as SFT but incorporates domain-specific rules into the prompt only at infer-

ence time to guide predictions.

The main contributions of this paper are twofold: a comparative evaluation of RAG, SFT, and RAG-FT for Islamic inheritance reasoning, and an empirical study across multiple open-source LLMs on the QIAS 2026 benchmark.

2. Background

2.1. Task Definition

The QIAS 2026 shared task focuses on Islamic inheritance reasoning, where the goal is to automatically solve inheritance scenarios described in natural language. Each scenario is presented in Arabic, specifying heirs and their relationships to the deceased (e.g.,

مات وترك: بنت ابن وأب الأب وأم الأب وأب أب الأب
(وابنان وأم وأربعة إخوة لأم. ما هو نصيب كل وريث؟

Given such input, the system must interpret the scenario, apply Islamic inheritance rules, and generate a structured JSON output. The expected output comprises multiple components: eligible heirs and their counts, blocked heirs, legal shares, whether awl or radd applies, the estate origin (asl al-mas'ala), and the final distribution of shares following ta'sil and adjustment when necessary.

Overall, the task combines natural language understanding, rule-based reasoning, and numerical computation, requiring models to perform multi-step decision-making in accordance with Islamic inheritance law.

2.2. Dataset Description

The dataset provided for the shared task (Boucekif et al., 2026) consists of 12,000 training instances and 500 test instances, with 200 training instances reserved for development. Each instance includes a unique identifier, a natural-language question describing an inheritance scenario, a reasoning trace, a category label, and a structured JSON output.

The dataset exhibits a clear class imbalance, with *awl* and *radd* cases underrepresented compared to simple cases. As reported in (Boucekif et al., 2026), *awl* and *radd* account for 4.81% ($n=577$) and 2.87% ($n=344$), respectively, in the training set, and 7.80% ($n=39$) and 1.00% ($n=5$) in the test set. In this work, we use the dataset as provided, without applying resampling or rebalancing strategies.

2.3. Related Work

Early work on Islamic inheritance focused on rule-based and expert systems for automating share calculations (Kurdi, 2026). With the advent of generative AI, there has been growing interest in evaluating its understanding of Islamic inheritance. Initial studies (Boucekif et al., 2025b; Justanieah and Kurdi, 2025) have explored this capability using multiple-choice questions (MCQs). For example, (Justanieah and Kurdi, 2025) evaluated ChatGPT on a limited set of undergraduate-level exam MCQs using a zero-shot prompting setup. The questions included both general knowledge of Islamic inheritance and scenario-based cases; however, due to the nature of MCQs, they typically focused on isolated aspects of inheritance problems, such as determining the share of a specific heir or identifying blocked heirs, rather than requiring complete end-to-end solutions. The study reported an accuracy of 65.35%, concluding that although ChatGPT demonstrated useful explanatory and reasoning abilities, it lacked the precision required for problems that demand both correct juridical interpretation and accurate mathematical computation.

Recent work has also emphasised the need for robust evaluation frameworks. (Kurdi and Justanieah, 2025) introduced a structured benchmark for inheritance calculation systems, showing that previously reported high-performing systems may perform significantly worse on more comprehensive cases. This emphasis on systematic evaluation was further extended in the QIAS 2025 shared task on Islamic Inheritance Reasoning and Islamic Knowledge Assessment (Boucekif et al., 2025a), which provided a large-scale benchmark for evaluating language models on inheritance and broader Islamic knowledge. Inheritance reasoning in QIAS 2025 was also formulated as an MCQ answering task, enabling systematic comparison across participating systems. The task demonstrated that

strong performance can be achieved using a variety of strategies, including fine-tuning (Elrefai et al., 2025), RAG (Hamed et al., 2025), combining fine-tuning with RAG (Mohammad, 2025), and ensemble-based approaches (AIDahoul and Zaki, 2025). Although QIAS 2025 enabled systematic comparison, it remained limited to answer selection rather than full end-to-end inheritance reasoning. To address this, QIAS 2026 introduces a more challenging setting in which systems are required to generate structured outputs that reflect the complete reasoning process.

3. System Overview

We explore three approaches for Islamic inheritance reasoning: 1) RAG-P, 2) SFT, and 3) RAG-FT. All approaches are built on top of pre-trained generative large language models. These approaches are described in the following sections, while implementation details, including prompts and the curated rule base, are publicly available at: <https://github.com/grkurdi/silah-inheritance-reasoning-llm>.

3.1. Base Models

We experimented with multiple open-source models, varying in size, architecture, and training paradigms, to assess their effectiveness for Islamic inheritance reasoning. Specifically, we evaluated four model families: Qwen: *Qwen2.5-3B-Instruct* (Yang et al., 2024), LLaMA: *Meta-Llama-3-8B-Instruct* (Grattafiori et al., 2024), DeepSeek: *DeepSeek-R1-Distill-Qwen-7B* (DeepSeek-AI, 2025), and Fanar: *Fanar-1-9B-Instruct* (Abbas et al., 2025). These models provide a balance between general-purpose reasoning and Arabic-focused modeling, with *Fanar* optimised for Arabic, while Qwen2.5, LLaMA, and DeepSeek offer strong general reasoning performance.

3.2. Multi-Stage Retrieval-Augmented Rule-Guided Pipeline

We propose a multi-stage, rule-guided approach for Islamic inheritance problem solving. Rather than generating the final answer in a single step, the pipeline decomposes the task into five stages: 1) special-case detection, 2) identification of heirs, blocked heirs, heir types, and initial shares, 3) computation of the estate origin and detection of broken groups in which shares cannot be evenly distributed among group members without fractions, 4) correction and determination of *awl* or *radd*, and 5) final adjustment as needed.

First, the question is preprocessed by extracting and normalising heir mentions. Second, relevant

rules are retrieved from a manually curated rule base. Third, a stage-specific prompt is constructed and sent to the model, incorporating task instructions, the retrieved rules, and, when applicable, the structured output of the previous stage. This sequential design improves control, interpretability, and alignment with the procedural nature of Islamic inheritance reasoning.

The rule base is derived from educational materials, primarily undergraduate lecture slides prepared by the authors. The rules are organised into JSON files by category, including special cases (22), fixed shares (26), asaba (11), blocking (22), asl al-mas'ala (7), adjustment (4), and awl/radd (13). Examples are provided in the Appendix.

Rule retrieval is performed in a stage-specific manner, where each stage selects a subset of rules tailored to its objective. Retrieval is based on the fields *heirs*, *required_heirs*, *optional_heirs_any*, and *conditions*. For example, in the special-case detection stage, a rule is selected if all *required_heirs* are present in the question. If *optional_heirs_any* is non-empty, at least one of its heirs must also be present. If both fields are empty, the rule is selected if any heir listed in *heirs* appears in the question.

3.3. Supervised Fine-tuning

We fine-tune a pretrained language model using parameter-efficient fine-tuning (PEFT) with Low-Rank Adaptation (LoRA) (Hu et al., 2022). LoRA adapters are applied to key projection layers (e.g., query, key, value, and feed-forward projections), allowing efficient adaptation with a reduced number of trainable parameters. Fine-tuning configuration and training settings are provided in the Appendix.

At inference, the model generates textual outputs that are parsed into structured JSON. Post-processing ensures schema consistency through field normalisation and structure validation. Specifically, normalisation functions enforce correct data types (e.g., integers, floats), standardise list formats (e.g., heirs and shares), and handle malformed or partial outputs. A final validation step ensures that each prediction conforms to the required schema before inclusion in the results.

3.4. Retrieval-Augmented Fine-Tuning

In addition to baseline fine-tuning, we incorporate a lightweight retrieval component that provides explicit inheritance rules at inference time. This approach follows the same supervised fine-tuning procedure as SFT and does not include retrieved rules during training; instead, retrieval is applied only at inference to guide predictions. We reuse the same curated rule set as in RAG-P for consistency.

The approach introduces three components. First, a preprocessing layer extracts and normalises heir mentions, decomposing compound expressions and standardising lexical variants into canonical singular forms. Second, a retrieval module selects relevant rules based on detected heirs by matching against rule metadata (e.g., *heirs*, *required_heirs*, *optional_heirs_any*) and returning a deduplicated set of rules. Third, instead of prompting the model with only task instructions and the question, the RAG-FT variant prepends the retrieved rules to the prompt.

4. Experimental Setup

4.1. Data Splits

We follow the official data splits. The models are trained on the training set, with the validation set used for parameter tuning and preliminary evaluation. The final results are reported on the test set, which is used solely for benchmarking.

4.2. Evaluation Metrics

The official evaluation metric is the MIR-E score, which ranges from 0 to 1 and is calculated as a weighted sum of stage-level scores. The weights are defined as $\lambda_1 = 0.30$ for eligible and blocked heirs, $\lambda_2 = 0.20$ for legal shares, $\lambda_3 = 0.10$ for the awl/radd decision, and $\lambda_4 = 0.20$ for the final distribution:

$$\text{MIR-E} = \lambda_1 S_{\text{heirs}} + \lambda_2 S_{\text{shares}} + \lambda_3 S_{\text{awl}} + \lambda_4 S_{\text{final}}$$

Each stage is evaluated according to its output type. Set-valued stages (eligible and blocked heirs) are assessed using set overlap, independent of order. The awl/radd stage, where the model determines whether the case involves awl, radd, or neither, is considered correct only if the predicted category matches the gold label and previous stages are correct. The numerical stages (legal shares and final distribution) are evaluated using a tolerance-based criterion, where the predictions are considered correct if they fall within 0.1 of the gold values.

5. Results

Our best result achieved seventh place on the leaderboard¹ with an MIR-E score of 0.83, demonstrating competitive performance compared to other submissions, where scores ranged from 0.74 to 0.94. Table 1 presents the performance of different models and approaches on the test set.

RAG-P achieved low performance across all models, with scores ranging from 0.17 to 0.38, highlighting limitations in retrieval precision and its im-

¹<https://sites.google.com/view/qias2026/leaderboards>

fact on reasoning. In particular, the retrieval step often returns multiple applicable rules for the same heir without sufficiently disambiguating the context. For example, a daughter may receive a fixed share of 1/2 or 2/3 of the estate, or, when combined with a son or a sister of the deceased, become a residuary heir and inherit the remainder after fixed shares are allocated, depending on the case. However, rules corresponding to all these scenarios are often retrieved simultaneously, introducing ambiguity and potentially confusing the model during inference.

Within this multi-stage RAG setting, performance varied across models. Fanar achieved the best results, while DeepSeek obtained the lowest score. A closer analysis revealed that DeepSeek often extracted only heirs, while other key components such as blocked heirs and the post-tasil distribution in the JSON output were frequently missing. This behaviour can be attributed to inconsistent adherence to the prompt instructions, which require Arabic-only responses and a strict output format. The model occasionally generated non-Arabic content (e.g., Chinese tokens) and deviated from the required structure, negatively affecting its results.

Based on these findings, we selected the best-performing models for fine-tuning. Fine-tuning significantly improved performance, with LLaMA and Fanar achieving MIR-E scores of 0.81 and 0.83, respectively. For the RAG-FT configuration, we used the best-performing base model, Fanar, which achieved a score of 0.74. These results highlight the effectiveness of fine-tuning over RAG-based approaches for Islamic inheritance reasoning, particularly in capturing structured rule application and producing accurate final distributions.

The best results were achieved using supervised fine-tuning, while RAG-P performed significantly worse, and RAG-FT yielded intermediate performance. Both RAG-P and RAG-FT introduced noise due to limitations in retrieval precision, as the retrieved rules were sometimes excessive and, in other cases, incomplete. Therefore, careful design of the retrieval mechanism is essential to ensure the selection of minimal, precise, and contextually relevant rules that capture interactions among combinations of heirs, rather than rules triggered solely by the presence of individual heirs.

Base Model	Approach	MIR-E
Qwen2.5	RAG-P	0.28
LLaMA	RAG-P	0.33
DeepSeek	RAG-P	0.17
Fanar	RAG-P	0.38
LLaMA	SFT	0.81
Fanar	SFT	0.83
Fanar	RAG-FT	0.74

Table 1: Model performance on the test set.

6. Limitations

This study has several limitations. First, although we consider multiple open-source models, we do not systematically examine the impact of model scale. Larger variants may exhibit different capabilities and achieve better performance. For example, we evaluate only *Qwen2.5-3B-Instruct*, as larger variants were excluded due to computational constraints. Future work should explore larger variants to enable a more comprehensive comparison.

Second, the absence of a pre-fine-tuning baseline limits our ability to disentangle the contribution of fine-tuning from the inherent capabilities of the models. However, results from team PSL (Mouhoub and BOUCHEKIF, 2026), who evaluate general-purpose models under controlled prompting without task-specific adaptation, provide a useful reference for estimating baseline capabilities.

Third, the comparison between fine-tuning and RAG-based approaches should be interpreted with caution, as our findings depend on the specific retrieval strategy and rules used. More advanced retrieval pipelines have achieved stronger results (Swaileh et al., 2026), suggesting that our RAG performance is constrained by retrieval quality and rule design. Moreover, we did not conduct a dedicated retrieval-level evaluation (e.g., precision and recall of retrieved rules), so attributing errors to retrieval noise remains a hypothesis requiring further investigation. A systematic analysis of retrieval quality and its impact on downstream generation will be explored in future work.

Finally, the analysis is limited to aggregate MIR-E scores, which do not fully capture the strengths and weaknesses of the proposed approaches. As future work, we plan to incorporate stage-level MIR-E breakdowns and conduct a detailed error analysis, thereby providing deeper insights for improving inheritance reasoning systems.

7. Conclusion

In this paper, we present our submission to the QIAS 2026 shared task. We evaluate three approaches based on open-source large language models: RAG-P, SFT, and RAG-FT. The best performance is achieved by SFT, which outperforms both RAG-based variants, demonstrating the effectiveness of learning inheritance reasoning patterns directly from data. In future work, we plan to explore training strategies that explicitly model rule identification and application as intermediate steps, enabling more accurate and interpretable reasoning. We also aim to improve adherence to instructions and output structure by investigating techniques such as constrained decoding.

8. Acknowledgments

We would like to thank the QIAS 2026 organisers for providing the datasets and organising this challenge.

9. Bibliographical References

- Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, Mus'ab Husaini, Soon-Gyo Jung, Ji Kim Lucas, Walid Magdy, Safa Messaoud, Abubakr Mohamed, Tasnim Mohiuddin, Basel Mousi, Hamdy Mubarak, Ahmad Musleh, Zan Naeem, Mourad Ouzzani, Dorde Popovic, Amin Sadeghi, Husrev Taha Sencar, Mohammed Shinoy, Omar Sinan, Yifan Zhang, Ahmed Ali, Yassine El Kheir, Xiaosong Ma, and Chaoyi Ruan. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.
- Nouar AIDahoul and Yasir Zaki. 2025. Nyuad at qias shared task: Benchmarking the legal reasoning of llms in arabic islamic inheritance cases. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 861–866.
- Abdessalam Boucekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026. [MAWARITH: A dataset and benchmark for legal inheritance reasoning with LLMs](#). *arXiv preprint arXiv:2603.07539*.
- Abdessalam Boucekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025a. [QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860.
- Abdessalam Boucekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al-Khatib, and Mohammed Ghaly. 2025b. [Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation](#). In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 246–257.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#).
- Eman Elrefai, Mohamed Lotfy Elrefai, and Aml Hassan Esmail. 2025. Gumball at qias 2025: Arabic llm automated reasoning in islamic inheritance. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 953–959.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Mohamed Motasim Hamed, Nada Ghneim, and Riad Sonbol. 2025. Hiastr at qias 2025: Retrieval-augmented llms with top-hit web evidence for arabic islamic reasoning qa. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 883–891.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Hala Justanieh and Ghader Kurdi. 2025. An evaluation of chatgpt's responses to islamic inheritance law questions: Insights and challenges. In *Proceedings of the Conference on Artificial Intelligence and Driving Change in Education*, Saudi Arabia.
- Ghader Reda Kurdi. 2026. A systematic mapping study: Exploring islamic inheritance in computing research. *International Journal of Informatics and Communication Technology (IJ-ICT)*. Accepted for publication (in press).
- Ghader Reda Kurdi and Hala Mohammad Justanieh. 2025. Enhancing evaluation practices for islamic inheritance calculation systems: Toward a standardized benchmark. *Indonesian Journal of Electrical Engineering and Computer Science*, 40(3):1549–1566.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- AL-Smadi Mohammad. 2025. Qu-nlp at qias 2025 shared task: A two-phase llm fine-tuning and retrieval-augmented generation approach for islamic inheritance reasoning. In *Proceedings of*

The Third Arabic Natural Language Processing Conference: Shared Tasks, pages 892–898.

Mohamed Lamine Mouhoub and Chahinez BOUCHEKIF. 2026. Psl at qias 2026: Which models perform better in arabic inheritance reasoning? In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*, Palma, Spain.

Wassim Swaileh, Mohammed-En-Nadhir Zighem, Hichem Telli, Salah Eddine Bekhouche, Abdelah Zakaria Sellam, and Fadi Dornaika. 2026. Cvpd at qias 2026: Rag-guided llm reasoning for al-mawarith share calculation and heir allocation. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*, Palma, Spain.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

10. Appendices

10.1. Examples of Inheritance Rules

Example 1:

```
{
  "id": "share_01",
  "title":
    "أصحاب الفروض (الزوجة)",
  "text":
    "الزوجة من أصحاب الفروض.",
  "category": "fixed_share",
  "madhhab": "general",
  "heirs": [
    "زوجة"
  ],
  "conditions": [],
  "lang": "ar"
}
```

Example 2:

```
{
  "id": "special_case_solution_03",
  "title":
    "حل المسألة الأكدرية",
  "text":
    "في هذه المسألة: ١) الزوج يرث النصف عند عدم وجود فرع وارث. ٢) الأم ترث الثلث بشرط عدم الفرع الوارث وعدم تعدد الإخوة وألا تكون المسألة من العمريتين. ٣) عند اجتماع الجد مع الأخت الشقيقة مع وجود صاحب فرض: يُنظر للجد في أفضل الأحوال من (المقاسمة، ثلث الباقي، سدس الكل)، وفي هذه المسألة سُدس الكل أفضل له فيأخذ السدس. ٤) تُضم الأخت الشقيقة أو الأخت لأب إلى الجد، فيما بقي ويُقسم بينهما للذكر مثل حظ الأنثيين.",
  "category": "special_case_solution",
  "madhhab": "general",
  "heirs":
    ["أخت لأب", "أخت شقيقة", "جد", "أم", "زوج"],
  "required_heirs":
    ["جد", "أم", "زوج"],
  "optional_heirs_any":
    ["أخت لأب", "أخت شقيقة"],
  "conditions": {},
  "lang": "ar"
}
```

10.2. Fine-Tuning Configuration and Training Details

All experiments were implemented using the HuggingFace Transformers library and PyTorch, and executed with GPU acceleration. The training configurations for LoRA and model optimisation are summarised in Tables 2 and 3.

Parameter	Value
Task type	CAUSAL_LM
LoRA rank (r)	16
LoRA alpha	32
LoRA dropout	0.05
Bias	none
Target modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj

Table 2: LoRA configuration.

Parameter	Value
Tokeniser	Same as base model tokeniser
Max sequence length	2048
Optimiser	AdamW
Learning rate	2×10^{-4}
Weight decay	0.01
Learning rate scheduler	Linear (no warmup)
Number of epochs	1
Train batch size per device	4
Eval batch size per device	1
Gradient accumulation steps	16
Effective train batch size	64
Selection metric	eval_loss
Greater is better	False
Mixed precision	FP16 when CUDA is available
Gradient checkpointing	Enabled
Use cache	Disabled
Seed	42

Table 3: Training configuration.

KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization

**Asma Al Wazrah¹, Waad Alshammari¹, Rawan Almatham¹, Raghad Al-Rasheed¹,
Afrah Altamimi¹, Rufael Marew³, Sawsan Alqahtani², Hanan Aldarmaki³,
Abdullah Alharbi¹, Abdulrahman Alshehri¹, Mohamed Assar¹, Amal Almazrui¹,
Abdulrahman AIOsaimy¹**

¹King Salman Global Academy for Arabic Language (KSAA), ²Princess Nourah bint Abdulrahman University (PNU), ³Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

^{1,2}Riyadh, Saudi Arabia, ³Abu Dhabi, United Arab Emirates

¹{aalwazrah, walshammari, ralmatham, a.altamimi, ralasheed, alshehri, masar, aalmazrui, aialharbi, aalosaimy}@ksaa.gov.sa, ²saalqhtani@pnu.edu.sa, ³{hanan.aldarmaki, rufael.marew}@mbzuai.ac.ae

Abstract

This paper presents the KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization, addressing a persistent challenge in Arabic NLP. The task focuses on transforming speech transcripts into fully diacritized Arabic text by leveraging both the speech signal and its undiacritized transcript. Unlike conventional ASR tasks that focus on transcription, this task integrates acoustic and textual information to improve diacritization accuracy. The shared task consists of two subtasks: (1) Data Contribution, where participants recorded and reviewed speech data through the VoiceWall platform, resulting in 2,160 recordings, and (2) Diacritization, where 5 teams developed systems that generate fully diacritized text from speech and undiacritized transcripts. The dataset includes approximately 5 hours of Modern Standard Arabic (MSA) and multi-dialectal speech with fully diacritized references. Experimental results show that several participant systems outperform the provided baselines, and that incorporating speech information and fine-tuning improves performance compared to text-only approaches. KSAA-2026 shared task establishes a benchmark for multimodal Arabic diacritization and supports the development of robust systems for applications in education, accessibility, and speech-driven text generation.

Keywords: Arabic Diacritization, Speech Dictation, Automatic Speech Recognition

1. Introduction

Arabic poses persistent challenges for NLP and speech processing due to the absence of diacritics in standard text and the wide dialectal diversity across the Arab world. The omission of diacritics introduces significant lexical and syntactic ambiguity, while dialectal variation adds phonological and lexical complexity that further complicates speech transcription.

Text-only diacritization approaches often struggle when applied to speech transcripts, as they fail to capture prosodic and phonetic cues present in the audio signal. At the same time, automatic speech recognition (ASR) systems rarely produce high-quality diacritized outputs (Shatnawi et al., 2024).

To address these challenges, KSAA-2026 shared task introduces a benchmark that integrates both speech signals and undiacritized transcripts as input, encouraging the development of multimodal approaches capable of generating fully diacritized text.

Recent studies further highlight these limitations. Shatnawi et al., (2024) show that text-only models suffer from domain and style mismatches when applied to ASR outputs, while speech-informed approaches lead to significant reductions in error rates. Similarly, Elgamal et al., (2024) emphasize the importance of diacritics in diverse real-world settings, demonstrating that even partial diacritization provides useful signals for improving model performance. The complexity of the task is

further illustrated in specialized domains; Alyafeai et al., (2023) report persistent challenges in Arabic poetry, where even state-of-the-art systems exhibit high diacritic error rates.

Motivated by these findings, this competition introduces a community-driven benchmark supported by continuous speech contributions, broader dialectal and speaker coverage, and multimodal modeling approaches. The expected impact spans multiple applications, including educational tools for Arabic learners, speech-driven writing assistants, and improved performance in downstream NLP tasks such as text-to-speech, machine translation, and parsing.

2. Dataset

To encourage community engagement and ensure sustainability, this shared task also includes a data contribution component, which is described in Section 3.1.

The core dataset for the competition consists of approximately 5 hours of Arabic speech audio collected from male and female speakers. The recordings represent a diverse range of nationalities, including Saudi, Egyptian, Kuwaiti, Bahraini, Sudanese, Qatari, Algerian, Syrian, and Palestinian speakers. This diversity ensures broad dialectal coverage alongside Modern Standard Arabic (MSA), making the dataset representative of real-world linguistic variation.

The dataset spans multiple domains such as politics, economy, sports, religion, and news. Each utterance is relatively short, not exceeding nine seconds, to facilitate alignment and model training.

Speech recordings are paired with corresponding transcripts and fully diacritized references. Basic preprocessing steps, such as normalization and alignment, were applied to ensure data usability.

The dataset is divided into training, development, and test sets as shown in Table 1.

Data split	No. of audio files	Duration
Train	2,327	~4.5 hours
Dev	260	~30 minutes
Test	329	~40 minutes

Table 1: Example of input and output structure.

The training and development sets were released to participants, while the test set was kept hidden to ensure a fair evaluation.

3. Task Description

The shared task is organized into two complementary subtasks designed to address both data collection and model development. The first subtask focuses on expanding the dataset through community contributions, while the second subtask evaluates system performance on automatic diacritization of speech dictation.

3.1 Task-1: Data Contribution

This subtask aims to enrich the shared dataset through open community participation. Anyone can join this track, and participants are invited to record Arabic speech via a dedicated web-based platform (VoiceWall)¹ using diacritized text prompts covering both Modern Standard Arabic (MSA) and a range of dialects.

Participants contribute in two complementary roles:

- **Record (Speak & Contribute):** Participants record speech samples by reading the provided diacritized prompts.
- **Listen and Review:** Participants listen to and review recordings submitted by others to support basic validation and filtering.

The validation process relies on two mechanisms: automated platform checks (e.g., verifying recording completeness, duration, and minimum audio quality thresholds) and cross-team review, whereby each team reviewed and rated the recordings submitted by another team. A recording is considered valid if it passes the automated checks and receives approval from the

reviewing team. This structured peer validation ensures a consistent and reproducible quality control process while maintaining broad participation.

3.2 Task-2: Automatic Diacritization of Speech Dictation

Participants in this task are required to build systems that generate fully diacritized Arabic text from speech dictation. Each input consists of a speech signal along with its corresponding undiacritized transcript, and the system output is expected to be a fully diacritized version of the transcript.

The task requires predicting full Arabic diacritics at the character level, including fatha, damma, kasra, sukūn, shaddah, and tanwīn marks for each character in the undiacritized text.

Two input settings are considered:

- **Text-only diacritization:** systems receive only the undiacritized transcript and generate the diacritized output.
- **Speech + text diacritization (main track):** systems leverage both the audio signal and the transcript to produce fully diacritized text.

A sample of the input–output data structure (see Table 2) is provided to illustrate the task format. This subtask establishes a benchmark for multimodal diacritization and supports the development of robust tools for education, accessibility, Quranic recitation, poetry, and dialectal speech processing.

Input	Speech Clip	
	Undiacritized transcript	أريد أن أشرب كوبًا من الشاي
Output	Diacritized transcript	أريدُ أنْ أَشْرَبَ كُوبًا مِنْ الشَّاي

Table 2: Example of input and output structure.

4. Evaluation

The evaluation framework is designed to assess both the data contribution efforts and the performance of diacritization systems. Each subtask is evaluated based on its specific objectives and criteria.

4.1 Task-1: Data Contribution Evaluation

The data contribution is assessed based on participant engagement and the amount of collected speech data. Each participant is

¹ <https://falak.ksaa.gov.sa/voicewall/ar/s/falak>

assigned 360 text prompts to record using the VoiceWall platform, covering Modern Standard Arabic (MSA) and various dialects.

To support data usability, the platform provides basic validation mechanisms, including listening and review functionalities that allow participants to check recordings and filter out unusable samples. The process is designed to encourage broad participation while maintaining an acceptable level of data quality.

Each submitted recording is associated with both the participant and the contributing team to ensure attribution and transparency. The collected data is aggregated and made available to participants for benchmarking purposes.

4.2 Task-2: Automatic Diacritization of Speech Dictation Evaluation

The submitted systems are evaluated based on their ability to generate fully diacritized Arabic text from speech and undiacritized transcripts.

Evaluation is conducted using the diac evaluation framework (Fadel style)², which computes Diacritic Error Rate (DER), Word Error Rate (WER), and Sentence Error Rate (SER), where lower values indicate better performance.

The evaluation is performed under two complementary dimensions. The first dimension considers whether case endings are included or excluded, distinguishing between full syntactic diacritization and core lexical diacritization. The second dimension considers whether characters with no diacritics are included or excluded from evaluation.

5. Systems

5.1 Baseline

We provide three baseline systems corresponding to the task setup. These baselines are intended as reference implementations to illustrate the task and are not optimized for performance.

- **Text-only baseline:** A transformer-based diacritization model trained only on the Tashkeela corpus and operating solely on undiacritized text.
- **Speech + text baseline:** A transformer-based model trained on Tashkeela and CartTTS, replicating the setup of Shatnawi et al., (2024), and using ASR outputs together with undiacritized text, without explicitly incorporating acoustic features.
- **Fine-tuned baseline:** A transformer-based model initialized from the previous setup and further fine-tuned on the training portion of the dataset in the challenge to improve diacritization performance.

5.2 Participating Systems

A total of 29 teams registered for Task-2, of which 5 teams successfully submitted valid results. The participating teams adopted a variety of approaches for the diacritization task. Below is a brief summary of the submitted systems.

Fine-Tashkeel team presents a character-level sequence-to-sequence approach that converts undiacritized text directly into fully diacritized output. The authors do not rely on task-specific fine-tuning for their final submission; instead, they evaluate 18 different model configurations, including text-only, ASR-based, and fine-tuned systems, and select the strongest one. Their analysis shows that text-only Seq2Seq models perform better than several multimodal systems that depend on off-the-shelf ASR. In addition to reporting leaderboard performance, the paper includes detailed error analysis across dialects and shows that dialectal variation and case endings remain the main sources of difficulty.

Eraserhead team presents a speech-aware diacritization pipeline built on a pretrained Transformer model with two inputs: text and ASR output. The main contribution is ASR consistency filtering, where noisy ASR transcripts are not removed but replaced with fallback text when they are judged unreliable, which makes training more stable. The authors train multiple runs with different filtering thresholds, combine them using confidence-based ensemble decoding, and then apply speaker-adaptive post-processing, which adjusts word-final diacritics based on each speaker's estimated style: speakers are classified as SUKUN, VOWEL, or MIXED based on development-set observations, and predictions are adjusted accordingly to reflect speaker-specific word-final diacritic behavior. They also add constrained decoding to ensure that the final output remains aligned with the original input words.

Abjad AI team proposed a multimodal system that extends a character-level Transformer by injecting speech representations through a lightweight fusion method called grouped speech conditioning, which is a mechanism that compresses the speech encoder's frame-level outputs into a small fixed number of pooled tokens (G tokens), projects them into the text representation space, and concatenates them with the text input before feeding the combined sequence into the diacritization model. Instead of using expensive fusion mechanisms such as cross-attention, this approach controls how much speech information is injected while keeping computational costs low. The authors compare different speech backbones, including several Whisper variants and Squeezeformer, and study the effect of varying the number of grouped

² <https://github.com/rufaelfekadu/Diac>

speech tokens ($G \in \{3, 5, 10, 15\}$). They also use a two-stage training schedule, first freezing the text model and then fine-tuning the whole system, and show that compact speech representations, especially Whisper-small with $G=5$, are the most effective in this setup.

Thaka team presents the winning system for Task-2, which builds on a multimodal CATT-Whisper architecture combining text and speech representations. Instead of modifying the model architecture, the authors focus on improving the training process using regularization techniques, including R-Drop, Focal Loss, and hyperparameter optimization. The speech encoder (Whisper) is kept frozen, while the text model is fine-tuned on the task data. At inference time, the system applies Monte Carlo Dropout with multiple stochastic forward passes and averages the predictions to improve robustness. The results show that training strategies and regularization play a critical role in improving diacritization performance in low-resource settings, achieving the best performance among all participating systems.

TantaArabNLP team presents a system based on the CATT-Whisper multimodal architecture, combining text and speech representations. The authors explore both text-only and multimodal models, with the best results achieved through fine-tuning the multimodal model on the official dataset. The approach focuses on selective fine-tuning and introduces a post-processing step to preserve punctuation, numbers, and non-Arabic tokens, addressing common errors in model outputs. Experimental results show clear improvements over baselines, achieving a WER

of 24.39% and securing second place in the competition.

These diverse approaches highlight different strategies for integrating textual and acoustic information, providing a solid basis for comparative evaluation.

6. Results

The results of the shared task are presented for both subtasks, covering data contribution and automatic diacritization. Each subtask is analyzed based on its specific objectives and evaluation criteria.

6.1 Task-1: Data Contribution Results

A total of 20 teams registered for Task-1, with 6 teams completing the data contribution process and actively submitting recordings through the platform. In total, 2,160 recordings were collected and reviewed. These contributions support the long-term goal of expanding Arabic speech resources. These contributions will be incorporated into future dataset expansions.

6.2 Task-2: Automatic Diacritization Results

The performance of all systems, including participant submissions and organizer baselines, is summarized in Table 3 under four evaluation settings. The primary metric (WER with case endings and including no-diacritic characters) is selected as it provides the most comprehensive and challenging evaluation, requiring accurate prediction of both diacritized and non-diacritized characters, as well as syntactic case endings; it therefore serves as the basis for overall system ranking.

Team	Including no diacritics						Excluding no diacritics					
	With case ending			Without case ending			With case ending			Without case ending		
	*DER	*WER	*SER	*DER	*WER	*SER	*DER	*WER	*SER	*DER	*WER	*SER
Thaka	6.87	23.26	66.16	5.97	17.05	53.96	6.64	18.64	64.33	4.80	10.48	50.91
TantaArabNLP	7.04	24.39	71.65	6.17	17.96	55.18	6.92	18.97	68.90	5.15	11.06	50.61
Abjad AI	7.51	25.34	73.48	6.60	18.66	60.06	7.51	19.95	70.73	5.68	11.85	55.49
Eraserhead	8.23	30.37	80.79	6.76	20.84	64.02	8.46	24.33	74.39	5.87	13.43	55.18
**Fine-tune text + ASR	9.91	31.84	82.93	7.89	20.99	67.07	8.52	24.73	78.66	4.82	10.89	50.61
Fine-Tashkeel	10.56	34.47	79.88	7.46	20.81	68.29	11.56	29.45	76.83	6.86	14.32	64.33
**text + ASR	13.50	40.24	82.32	10.58	27.95	71.95	14.26	33.03	75.61	9.96	19.71	60.37
**text only	17.66	49.85	91.77	13.23	32.24	82.62	20.08	46.20	91.77	13.93	27.07	81.71

*Lower is better

** Baseline

Table 3: Test set results for participant systems and organizer baselines under four evaluation settings.

Among the baselines, the fine-tuned model achieves the best performance, confirming the effectiveness of adapting models to the task-specific training data. However, four participant systems (Thaka, TantaArabNLP, Abjad AI, and Eraserhead) outperform the baselines across evaluation settings. In particular, these systems reduce WER from 31.84% (best baseline) to approximately 23–25% and DER from 9.91% to around 6–8%, demonstrating substantial gains.

A clear performance gap is observed between the text-only baseline and models incorporating ASR outputs, highlighting the importance of speech information for improving diacritization. Furthermore, fine-tuning yields notable improvements over the non-adapted speech+text baseline, especially in DER and WER, indicating better generalization to the target domain.

Across all systems, removing case endings significantly reduces error rates, confirming that syntactic diacritics remain one of the most challenging aspects of Arabic diacritization. Similarly, excluding no-diacritic characters leads to improved scores, suggesting that systems struggle to correctly predict when diacritics should be absent, in addition to predicting their correct forms.

7. Limitations

While KSAA-2026 represents a meaningful step toward multimodal Arabic diacritization, several limitations should be acknowledged. First, the dataset is relatively small in scale, comprising approximately five hours of speech, which may limit the generalizability of trained models to broader real-world conditions. Second, although the dataset covers nine nationalities and includes both Modern Standard Arabic and dialectal speech, the dialectal coverage remains uneven, with some varieties underrepresented relative to their speaker populations across the Arab world. Third, the data contribution track relied on cross-team peer review rather than expert linguistic annotation, which, while practical and scalable, may introduce inconsistencies in quality assessment. Fourth, the evaluation framework focuses on character- and word-level diacritic accuracy but does not capture downstream task performance, such as text-to-speech naturalness or reading comprehension support. Finally, the shared task setting constrains participants to the officially released training data, which limits exploration of transfer learning from larger external resources and may not reflect the full potential of current modeling approaches.

8. Conclusion

KSAA-2026 shared task presented a benchmark for Arabic speech diacritization, highlighting the challenges of combining speech and text for accurate diacritic prediction. The results show that leveraging acoustic information and fine-tuning significantly improves performance over baseline systems.

The diversity of submitted approaches demonstrates the potential for further advancements in multimodal Arabic NLP, while the data contribution track supports the continuous expansion of Arabic speech resources.

9. Ethics Statement

All speech data collected as part of KSAA-2026 was obtained with the informed consent of participants, who were made aware of how their recordings would be used for research purposes. Participation was voluntary, and no personally identifiable information beyond speaker nationality and gender was collected. Recordings were made using scripted prompts covering publicly available text domains, and the dataset is intended solely for non-commercial academic research. Speaker diversity across nationalities and dialects was intentionally incorporated to promote representational fairness. We acknowledge that diacritization systems may carry unintended biases related to dialect or speaking style, and encourage future work to assess and mitigate such biases.

10. References

- Alyafeai, Z. and Ahmad, I. (2023). Ashaar: Automatic analysis and generation of Arabic poetry using deep learning approaches. arXiv:2307.06218. doi:10.48550/arXiv.2307.06218.
- Elgamal, M., Torki, M., and Hussein, A. (2024). Arabic diacritics in the wild: Exploiting opportunities for improved diacritization. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL).
- Shatnawi, A., Alqahtani, S., and Aldarmaki, H. (2024). Automatic restoration of diacritics for speech data sets. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).

Thaka at KSAA-2026 Task 2: Regularized Fine-Tuning for Arabic Speech Diacritization

Meshal Alamr, Hassan Alqaeri, Abdullah Aldahlawi

Thaka, Advanced AI and Information Technology

Riyadh, Saudi Arabia

{m.alamr, h.alqaeril, aldahlawi}@thakaait.net

Abstract

We describe the winning system for Task 2 of the KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization (Al Wazrah et al., 2026). The task requires producing fully diacritized Arabic text from speech audio and undiacritized transcripts, with only 2,327 training samples available and no external data permitted. Our system fine-tunes CATT-Whisper (Ghannam et al., 2025), a character-level multimodal model combining a pretrained CATT text encoder with a frozen Whisper speech encoder. The key to our approach is training regularization: R-Drop consistency regularization, Optuna-optimized hyperparameters with high weight decay, and Focal Loss. At inference, we average 200 stochastic forward passes across four model checkpoints using Monte Carlo Dropout at the softmax probability level. The system achieves **23.26% WER** on the primary leaderboard metric (with case endings, including no-diacritic positions), placing 1st among all participants.

Keywords: Arabic diacritization, multimodal, speech processing, regularization, shared task, KSAA-2026

1. Introduction

Arabic text is typically written without diacritics, the short vowels and phonological markers that determine pronunciation and meaning. Restoring these marks automatically is important for text-to-speech, machine translation, and language learning (Habash and Rambow, 2007), but remains challenging: text-only models (Alasmay et al., 2024) struggle with inherent ambiguity, particularly in dialectal Arabic where phonetic variation is high. Speech signals offer a complementary source of disambiguation, as prosodic and phonetic cues can resolve cases that are irrecoverable from orthography alone.

The KSAA-2026 Shared Task (Al Wazrah et al., 2026) presents a low-resource instance of this problem: only 2,327 training samples across multiple Arabic dialects, with no external data permitted. CATT-Whisper (Ghannam et al., 2025) demonstrated that a multimodal architecture combining CATT and Whisper encoders is effective for this task, achieving strong results at NADI 2025. We adopt this architecture and focus on making the most of the limited training data through regularization and inference-time ensembling.

2. Background

2.1. Task and Data

Task 2 of the KSAA-2026 Shared Task requires participants to produce fully diacritized Arabic text given speech audio and undiacritized transcripts. Systems are evaluated using Diacritic Error Rate (DER), Word Error Rate (WER), and Sentence Er-

ror Rate (SER) (Fadel et al., 2019), with **WER** (with case endings, including no-diacritic positions) as the primary ranking metric. All results in this paper are reported under this setting.

The dataset (Al Wazrah et al., 2026) contains 2,327 training, 260 development, and 328 test samples spanning multiple Arabic dialects. Each sample is a short utterance (average ~7 seconds) paired with a diacritized transcript. We filter samples where the ratio of diacritized to total characters is below 0.6, yielding 2,187 effective training samples.

2.2. Related Work

Arabic diacritization has progressed from morphological taggers (Habash and Rambow, 2007) through transformer-based systems (Nazih and Hifny, 2022) to pretrained encoders such as AraBERT (Antoun et al., 2020) and the character-level CATT model (Alasmay et al., 2024). Ghannam et al. (2025) introduced CATT-Whisper, which fuses a CATT text encoder with Whisper (Radford et al., 2023) speech features via prefix addition, winning the NADI 2025 dialectal diacritization track. R-Drop (Liang et al., 2021) penalizes divergent predictions between two dropout-masked forward passes and has been effective in low-resource settings. MC Dropout (Gal and Ghahramani, 2016) enables ensemble-like inference from a single model by keeping dropout active at test time. Our work applies these regularization and inference techniques to multimodal Arabic diacritization.

Hyperparameter	Value
Learning rate	4.1×10^{-6}
R-Drop α	2.08
Focal γ / label smoothing	0.34 / 0.018
Weight decay	0.098
Speech emb. dropout	0.09
Batch size / epochs	16 / 40
Warmup / min LR factor	3 ep / 0.002
SpecAugment (freq / time)	10 / 63
Noise injection	Gauss., SNR 10–30 dB
Whisper unfrozen	0 (fully frozen)

Table 1: Hyperparameters selected via Optuna (30 trials).

3. System Overview

3.1. Architecture

Our system builds on CATT-Whisper. The text encoder is a 6-layer CATT Transformer ($d = 512$, 16 heads), pretrained on Arabic diacritization, predicting one of 15 diacritic classes per Arabic letter. The speech encoder is Whisper-base (Radford et al., 2023) (6 encoder blocks, $d = 512$), kept fully frozen in the primary configuration. Fusion uses prefix addition: 1,500 Whisper frames are mean-pooled into 150 tokens, projected, and added to 150 dedicated prefix positions that precede the text input. The model has ~ 39 M parameters (~ 19 M trainable).

3.2. Training

Figure 1(a) illustrates the training procedure. We fine-tune with R-Drop: each input passes through the model twice with different dropout masks, and a symmetric KL divergence penalty ($\alpha = 2.08$) encourages consistent predictions. The supervised loss is Focal Loss (Lin et al., 2017) ($\gamma = 0.34$) with label smoothing ($\epsilon = 0.018$). Speech embedding dropout ($p = 0.09$) randomly zeros the speech representation during training, following the modality-robust training scheme of CATT-Whisper. Audio augmentation uses SpecAugment (Park et al., 2019) and Gaussian noise injection. Hyperparameters were selected using Optuna (Akiba et al., 2019) (30 trials, 12 epochs each); Table 1 shows the final configuration. We use AdamW (Loshchilov and Hutter, 2019) with cosine learning rate decay. We train four checkpoints: three with this configuration using seeds 42, 7, and 123, and a fourth from a separate Optuna trial with an alternative configuration (learning rate 4.7×10^{-5} , batch size 32, $\gamma = 1.0$, label smoothing 0.108, and 4 unfrozen Whisper blocks after epoch 15) to increase ensemble diversity.

System	DER ↓	WER ↓	SER ↓
meshal (Ours)	6.87	23.26	66.16
nadaadelmousa	7.04	24.39	71.65
naif_alharthi	7.51	25.34	73.48
nahian_abu	8.23	30.37	80.79
Hassan	10.56	34.47	79.88
omarnj10	27.94	44.05	98.78
astral_fate	31.67	84.50	99.70
Baseline (FT text+ASR)	9.91	31.84	82.93
Baseline (text+ASR)	13.50	40.24	82.32
Baseline (text-only)	17.66	49.85	91.77

Table 2: Test set results. All metrics: with case endings, incl. no-diacritic positions. Ranked by WER (primary metric).

3.3. Inference

Figure 1(b) illustrates the inference procedure. At inference, we keep dropout active in the CATT encoder ($p = 0.1$) while LayerNorm stays in eval mode. Each of the four models runs 50 stochastic forward passes ($4 \times 50 = 200$ total), and we average softmax probabilities across all passes before taking the argmax.

3.4. Post-Processing

We use direct positional diacritic insertion: CATT models maintain 1:1 correspondence between Arabic letter positions and predicted diacritics. We enforce three invariants: (1) stripping diacritics from the output recovers the original input; (2) diacritic count matches predictions; (3) all letter positions are consumed.

4. Results

Table 2 shows the test set results compared to other participants and the shared task baselines. Our system achieves **23.26% WER** on the primary metric.

Table 3 presents a cumulative ablation on the development set, starting from the pretrained CATT-Whisper model and progressively adding our modifications. Fine-tuning CATT-Whisper with a standard recipe (cross-entropy loss, learning rate 10^{-5}) gives 30.43% WER; adding our regularized recipe (R-Drop, Focal Loss, high weight decay) reduces this to 27.18%, and MC Dropout ensembling brings it to 26.02%. The majority of the gain (3.25 pp) comes from the regularized training recipe; MC Dropout adds a further 1.16 pp, confirming that the training recipe is the primary driver of improvement.

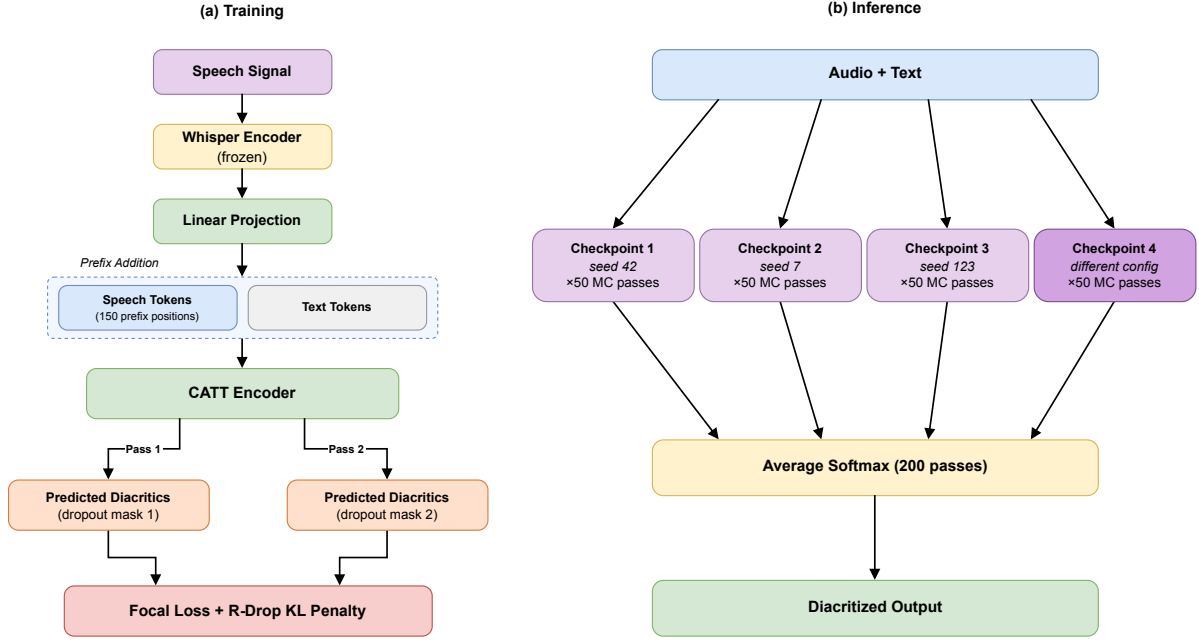


Figure 1: (a) Training: speech features from the frozen Whisper encoder are fused with text tokens via prefix addition and processed by the CATT encoder. R-Drop runs two forward passes with different dropout masks, optimizing with Focal Loss and a KL consistency penalty. (b) Inference: four checkpoints each run 50 MC Dropout passes; the 200 softmax distributions are averaged to produce the diacritized output.

Configuration	DER	WER
CATT-Whisper (pretrained)	17.76	54.06
CATT-Whisper (fine-tuned) [†]	8.59	30.43
+ Regularized recipe [‡]	7.57	27.18
+ 4-model MC Dropout ensemble	7.17	26.02

[†] Baseline: $\text{lr}=10^{-5}$, cross-entropy loss, batch size 16, 30 epochs.

[‡] R-Drop + Focal Loss + high weight decay (Optuna, 30 trials).

Table 3: Cumulative ablation on dev set (% , with case endings, incl. no-diacritic positions).

4.1. Discussion

Training recipe. We also explored architectural modifications including cross-attention fusion, CRF decoding, attention pooling, auxiliary heads, and RL fine-tuning, none of which improved over the fine-tuned CATT-Whisper baseline. The regularized recipe yielded a 3.25pp WER gain (Table 3), suggesting that, in our experiments with 2,187 training samples, the optimization strategy matters more than the model architecture. MC Dropout ensembling adds 1.16 pp at a cost of 200 forward passes ($\sim 50\times$ slower than a single pass); in practice, reducing the number of passes and studying the effect on accuracy could yield a more efficient trade-off.

Input	الظاهر أنه لا خلاف في الحقيقة للاتفاق على امتناع إدراك حقيقة الذات
Ours	الظاهر أنه لا خلاف في الحقيقة للاتفاق على امتناع إدراك حقيقة الذات
Gold	الظاهر أنه لا خلاف في الحقيقة للاتفاق على امتناع إدراك حقيقة الذات

Table 4: Dev set example.

Audio contribution. Ghannam et al. (2025) showed that incorporating speech features significantly improves diacritization accuracy. Consistent with their findings and the gap between the text-only and audio-equipped baselines in Table 2, we observed a similar pattern when fine-tuning CATT-Whisper without speech features, further confirming the importance of audio input for this task.

Qualitative example. Table 4 shows a dev set example where our system correctly diacritizes a complex sentence with case endings and shaddas.

5. Conclusion

We presented our system for KSAA-2026 Task 2, which fine-tunes CATT-Whisper with R-Drop, Focal Loss, and Optuna-optimized hyperparameters,

and uses MC Dropout ensembling at inference. At this data scale, training regularization yielded larger gains than any of the architectural modifications we explored. Future work could disentangle the individual contributions of each regularization component and examine per-dialect performance to identify remaining challenges.

Acknowledgements

We thank Thaka for supporting this work, KSAA for organizing the shared task, and the Abjad AI team for open-sourcing the CATT-Whisper model.

References

- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631.
- Asma Al Wazrah, Waad Alshammari, Rawan Almatham, Raghad Al-Rasheed, Afrah Altamimi, Rufael Marew, Sawsan Alqahtani, Hanan Aldarmaki, Abdullah Alharbi, Abdulrahman Alshehri, Mohamed Assar, Amal Almazrua, and Abdulrahman AlOsaimy. 2026. KSAA-2026 shared task on Arabic speech dictation with automatic diacritization. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*.
- Faris Alasmay, Orjuwan Zaafarani, and Ahmad Ghannam. 2024. [CATT: Character-based Arabic tashkeel transformer](#). In *Proceedings of the Second Arabic Natural Language Processing Conference (ArabicNLP 2024)*.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, pages 9–15.
- Ali Fadel, Ibraheem Tuffaha, Bara’ Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019. Arabic text diacritization using deep neural networks. *arXiv preprint arXiv:1905.01965*.
- Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059.
- Ahmad Ghannam, Naif Alharthi, Faris Alasmay, Kholood Al Tabash, Shouq Sadah, and Lahouari Ghouti. 2025. [Abjad AI at NADI 2025: CATT-Whisper: Multimodal diacritic restoration using text and speech representations](#). In *Proceedings of the Second Arabic Natural Language Processing Conference: Shared Tasks (ArabicNLP 2025)*.
- Nizar Habash and Owen Rambow. 2007. [Arabic diacritization through full morphological tagging](#). In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56.
- Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-Drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34.
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.
- Waleed Nazih and Yasser Hifny. 2022. [Arabic syntactic diacritics restoration using BERT models](#). *Computational Intelligence and Neuroscience*, 2022:3214255.
- Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, pages 2613–2617.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 28492–28518.

TantaArabNLP at KSAA-2026 Task 2: Adapting CATT-Whisper for Arabic Speech Dictation with Automatic Diacritization

Nada Esmaeil, Reda M. Elbasiony, Mohamed T. Faheem

Faculty of Engineering, Tanta University

Tanta, Egypt

{nada_adel, reda, mohamed_ahmed1}@f-eng.tanta.edu.eg

Abstract

We present our submission to the KSAA-2026 Shared Task (Subtask 2): Automatic Diacritization of Speech Dictation. Building upon the CATT-Whisper multimodal architecture, which fuses representations from a pre-trained CATT text encoder and the Whisper speech encoder, we fine-tune the model end-to-end on the official shared task training data. To further enhance performance on speech-dictated Arabic text, we apply careful post-processing to the model outputs. Our best submission achieves a Diacritic Error Rate (DER) of 7.04, a Word Error Rate (WER) of 24.39, and a Sentence Error Rate (SER) of 71.65 on the hidden test set, securing 2nd place in the competition. These results demonstrate the effectiveness of adapting a strong multimodal baseline to the speech-aware diacritization setting and highlight the value of task-specific fine-tuning and output refinement for bridging the gap between spoken transcripts and fully diacritized Arabic text.

Keywords: Arabic diacritization, Speech-aware diacritization, Multimodal diacritic restoration, CATT-Whisper

1. Introduction

The Arabic language is characterized by a rich diacritic system that includes short vowels (fathā, ḍamma, kasra), sukūn, shaddah, and tanwīn marks. These diacritics are essential for correct pronunciation, meaning, and grammatical structure. However, diacritics are routinely omitted in most written Arabic texts, including speech transcripts. This creates significant ambiguity for computational systems and downstream applications such as ASR, TTS, and machine translation.

Traditional text-only diacritization models perform well on formal MSA but struggle with speech transcripts, which are less formal and may include repetitions or colloquial elements. KSAA-2026 Subtask 2 introduces a *speech-aware diacritization* benchmark, requiring systems to combine raw audio and undiacritized transcripts to produce fully diacritized text.

In this work, we describe our submission to KSAA-2026 Subtask 2, building on the multimodal CATT-Whisper architecture (Ghannam et al., 2025). Our main contributions are:

- Fine-tuning CATT-Whisper Early Fusion on the official dataset using selective layer tuning with differential learning rates.
- A deterministic post-processing module to restore punctuation, numbers, and non-Arabic tokens.
- Achieving a DER of 7.04, WER of 24.39, and SER of 71.65 on the hidden test set, securing second place.

2. Related Work

2.1. Text-based Diacritic Restoration

Early Arabic diacritic restoration systems relied heavily on rule-based and morphological analyzers. Prominent examples include MADAMIRA (Pasha et al., 2014) and Camelira (Obeid et al., 2022), which perform diacritization as part of full morphological disambiguation. Subsequent research moved toward neural sequence labeling approaches using RNNs, BiLSTMs with CRFs (Althubaity et al., 2020), and transformer-based models. Among recent models, the Character-based Arabic Tashkeel Transformer (CATT) (Alasmay et al., 2024) has established itself as a strong character-level baseline for Arabic diacritization.

Despite notable progress on well-formed Modern Standard Arabic (MSA) text, text-only models often suffer from significant domain mismatch when applied to speech transcripts, which tend to be more informal, repetitive, or dialect-influenced (Aldarmaki and Ghannam, 2023).

2.2. Speech-aware and Multimodal Diacritic Restoration

The use of acoustic information to support diacritic restoration remains relatively underexplored. One of the earliest attempts dates back to Vergyri and Kirchhoff (Vergyri and Kirchhoff, 2004), who combined acoustic features with morphological analysis for dialectal Arabic. More recently, Aldarmaki and Ghannam (Aldarmaki and Ghannam, 2023) showed that text-based diacritizers generalize poorly to ASR outputs and that models trained

on gold diacritized speech data achieve superior performance.

With the advancement of large pretrained models, multimodal approaches have begun to emerge. Shatnawi et al. (Shatnawi et al., 2024) explored cascaded pipelines in which a fine-tuned Whisper model first generates diacritized transcripts that are then refined by a text-based restoration model.

The most relevant work to our submission is the CATT-Whisper architecture (Ghannam et al., 2025), which integrates a CATT text encoder with the Whisper speech encoder using both early fusion and cross-attention mechanisms. Originally developed for dialectal Arabic diacritic restoration in the NADI 2025 shared task, CATT-Whisper demonstrated the effectiveness of combining textual and acoustic signals in a multimodal framework.

3. Background

3.1. Task Setup

The KSAA-2026 Shared Task (Subtask 2), titled *Automatic Diacritization of Speech Dictation*, requires participants to develop multimodal systems that take as input a raw Arabic speech audio file together with its corresponding undiacritized transcript and generate a fully diacritized version of the transcript at the character level. The model must predict all Arabic diacritic marks, including fatha, damma, kasra, sukūn, shaddah, and tanwīn.

Systems are evaluated using three metrics: Diacritic Error Rate (DER), Word Error Rate (WER), and Sentence Error Rate (SER), with WER serving as the primary metric. Results are reported with and without case endings (i'rāb). Case endings constitute the most challenging aspect of Arabic diacritization as they heavily depend on syntactic context.

3.2. Dataset

The dataset was collected through the VoiceWall crowdsourcing platform developed by the King Salman Global Academy for Arabic Language (KSAA). It consists of approximately five hours of high-quality Arabic speech recordings from both male and female speakers, covering Modern Standard Arabic (MSA) as well as various Arabic dialects. All utterances are short (maximum 9 seconds) to ensure accurate speech-text alignment.

The official dataset splits are summarized in Table 1.

Split	# Files	Duration	Gender
Training	2,327	≈4.5 h	1,423M / 904F
Development	260	≈0.5 h	161M / 99F
Test	260	≈0.5 h	154M / 104F

Table 1: Statistics of the KSAA-2026 Subtask 2 dataset.

4. System Overview

We built our submission for KSAA-2026 Subtask 2 upon the CATT-Whisper multimodal architecture (Ghannam et al., 2025). In addition, we conducted experiments with the pure CATT model using encoder-decoder configuration. This section describes the architectural choices and the experiments we performed.

4.1. Baselines

The organizers provided three baseline systems as reference implementations:

- **Text-only:** A transformer model using only undiacritized text.
- **Speech + text:** A transformer model combining ASR outputs with undiacritized text, without raw acoustic features.
- **Fine-tuned:** A transformer model fine-tuned on the official KSAA-2026 training data.

4.2. Architectural Choices

4.2.1. CATT Text-only Model

We used the Encoder-Decoder (ED) variant of the pretrained CATT model (Alasmary et al., 2024). In this setup, the undiacritized Arabic text is fed as input and the model generates the full sequence of diacritics in an auto-regressive manner. This architecture conditions each predicted diacritic on both the input text and the previously generated diacritics, which is particularly helpful for modeling long-range dependencies.

4.2.2. CATT-Whisper Multimodal Model

Our best-performing system is based on the CATT-Whisper architecture (Ghannam et al., 2025). This model combines:

- A pretrained CATT model as the **text encoder**.
- The encoder part of the Whisper-Base model as the **speech encoder**.
- A linear projection layer to align the speech embeddings with the text embedding dimension.

5. Experimental Setup

We conducted experiments with two main systems: the CATT Encoder-Decoder model and the CATT-Whisper Early Fusion model. For both architectures, we first evaluated the official pre-trained checkpoints without any additional fine-tuning, followed by the application of our post-processing module. We then performed task-specific fine-tuning on the official KSAA-2026 training data.

5.1. Preprocessing

We followed the preprocessing pipeline used in the original CATT models (Alasmary et al., 2024), but set the Tashkeel-to-text ratio threshold to 0, retaining all samples due to the high quality and limited size of the KSAA-2026 training data.

5.2. Fine-tuning Setup

We fine-tuned both the CATT Encoder-Decoder and CATT-Whisper (Early Fusion) models on the official KSAA-2026 training split (2,327 utterances). Training was performed for a maximum of 20 epochs with early stopping based on validation DER. We used a batch size of 32 and a dropout rate of 0.01.

The AdamW optimizer was employed with layer-wise learning rates. We assigned a higher learning rate (1×10^{-3}) to the final classification head (and the speech projection head in CATT-Whisper) because these components are task-specific and benefit from faster adaptation. A much lower learning rate (4×10^{-5}) was used for the last layer of the encoder/decoder to gently update the pretrained knowledge without catastrophic forgetting. All remaining layers were frozen to preserve the rich representations learned during pretraining and to reduce computational cost and overfitting risk on the relatively small KSAA training set.

5.3. Post-processing

To mitigate the tendency of both CATT and CATT-Whisper models to drop or distort non-Arabic tokens (punctuation, numbers, symbols, and whitespace), we apply a deterministic post-processing step. The module preserves all formatting and non-Arabic characters from the original transcript while retaining the model’s predicted diacritics on Arabic letters. Specifically, newlines and whitespace are taken exclusively from the original input, Arabic base letters and their diacritics are taken from the model output, and all other characters are copied directly from the original. An illustrative example is shown in Table 2. This alignment-based restoration significantly improves consistency and final evaluation scores.

Example Input	وهي قد تخرج خطاطين (متقنين)
CATT-Whisper Output	وهي قد تخرج خطاطين متقنين
After Post-Processing	وهي قد تخرج خطاطين (متقنين)
Reference	وهي قد تخرج خطاطين (متقنين)

Table 2: Example from the KSAA-2026 development/test set showing CATT-Whisper output before and after applying the post-processing step, aligned with the reference.

6. Results

We evaluated our systems on both the development and hidden test sets of the KSAA-2026 Shared Task (Subtask 2). Tables 3, 4, and 5 present the Diacritic Error Rate (DER), Word Error Rate (WER), and Sentence Error Rate (SER) on the development set, respectively. Tables 6, 7, and 8 show the corresponding results on the hidden test set. All results are reported under two evaluation settings: with and without case-ending diacritics, and with and without including characters labeled as “no diacritic” in the evaluation.

In the zero-shot setting, CATT and CATT-Whisper achieve relatively similar performance. However, after fine-tuning on the KSAA-2026 training data, CATT-Whisper consistently outperforms CATT, particularly in WER (28.56 vs 33.50 on the development set). This demonstrates the benefit of combining acoustic features with task-specific adaptation.

On the hidden test set, our fine-tuned CATT-Whisper achieves the best performance with a DER of 7.04, WER of 24.39, and SER of 71.65. It outperforms all baselines, including the Fine-tuned Text+ASR baseline, confirming the advantage of multimodal early fusion. The post-processing step further improved the results by recovering punctuation and non-Arabic tokens frequently omitted by the base models.

7. Conclusion

In this paper, we presented our submission to the KSAA-2026 Shared Task (Subtask 2). We adapted the CATT-Whisper Early Fusion architecture and applied selective fine-tuning with differential learning rates on the official training data, followed by a post-processing module to restore non-Arabic tokens and punctuation. On the hidden test set, our best model achieved a DER of 7.04, WER of 24.39, and SER of 71.65, securing second place in the competition. These results demonstrate the effectiveness of multimodal approaches for speech-aware Arabic diacritization and highlight the importance of task-specific fine-tuning and output refinement.

Model	Including ‘no diacritic’		Excluding ‘no diacritic’	
	w. CE	w.o CE	w. CE	w.o CE
CATT (zero-shot)	13.08	7.02	15.62	8.07
Fine-tuned CATT	9.63	5.78	11.38	6.51
CATT-Whisper (zero-shot)	13.15	7.51	15.91	8.86
Fine-tuned CATT-Whisper	8.06	5.84	9.30	6.47
Text-only Baseline	20.97	15.13	23.29	15.21

Table 3: Diacritic Error Rate (DER) on the development set.

Model	Including ‘no diacritic’		Excluding ‘no diacritic’	
	w. CE	w.o CE	w. CE	w.o CE
CATT (zero-shot)	42.65	17.91	41.27	16.90
Fine-tuned CATT	33.50	15.14	32.04	13.87
CATT-Whisper (zero-shot)	42.95	19.55	41.79	18.58
Fine-tuned CATT-Whisper	28.56	16.07	26.54	14.21
Text-only Baseline	55.33	33.68	52.52	28.07

Table 4: Word Error Rate (WER) on the development set.

Model	Including ‘no diacritic’		Excluding ‘no diacritic’	
	w. CE	w.o CE	w. CE	w.o CE
CATT (zero-shot)	79.23	63.08	79.23	60.77
Fine-tuned CATT	78.46	57.31	76.54	54.62
CATT-Whisper (zero-shot)	82.31	68.46	82.31	66.92
Fine-tuned CATT-Whisper	81.15	63.46	79.23	58.46
Text-only Baseline	94.62	83.85	94.23	81.54

Table 5: Sentence Error Rate (SER) on the development set.

Model	Including ‘no diacritic’		Excluding ‘no diacritic’	
	w. CE	w.o CE	w. CE	w.o CE
Text-only Baseline	17.66	13.23	20.08	13.93
Text+ASR Baseline	13.50	10.58	14.26	9.96
Fine-tuned Text+ASR Baseline	9.91	7.89	8.52	4.82
Our Fine-tuned CATT-Whisper	7.04	6.17	6.92	5.15

Table 6: Diacritic Error Rate (DER) on the hidden test set.

Model	Including ‘no diacritic’		Excluding ‘no diacritic’	
	w. CE	w.o CE	w. CE	w.o CE
Text-only Baseline	49.85	32.24	46.20	27.07
Text+ASR Baseline	40.24	27.95	33.03	19.71
Fine-tuned Text+ASR Baseline	31.84	20.99	24.73	10.89
Our Fine-tuned CATT-Whisper	24.39	17.96	18.97	11.06

Table 7: Word Error Rate (WER) on the hidden test set.

Model	Including ‘no diacritic’		Excluding ‘no diacritic’	
	w. CE	w.o CE	w. CE	w.o CE
Text-only Baseline	91.77	82.62	91.77	81.71
Text+ASR Baseline	82.32	71.95	75.61	60.37
Fine-tuned Text+ASR Baseline	82.93	67.07	78.66	50.61
Our Fine-tuned CATT-Whisper	71.65	55.18	68.90	50.61

Table 8: Sentence Error Rate (SER) on the hidden test set.

8. Bibliographical References

- Abdulmohsen Al-thubaity, Atheer AlKhalifa, Abdulrahman Almuhareb, and Waleed Alsanie. 2020. [Arabic diacritization using bidirectional long short-term memory neural networks with conditional random fields](#). *IEEE Access*, PP:1–1.
- Faris Alasmary, Orjuwan Zaafarani, and Ahmad Ghannam. 2024. [CATT: Character-based Arabic tashkeel transformer](#). In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 250–257, Bangkok, Thailand. Association for Computational Linguistics.
- Hanan Aldarmaki and Ahmad Ghannam. 2023. [Diacritic recognition performance in arabic asr](#). pages 361–365.
- Ahmad Ghannam, Naif Alharthi, Faris Alasmary, Kholood Al Tabash, Shouq Sadah, and Lahouari Ghouti. 2025. [Abjad AI at NADI 2025: CATT-whisper: Multimodal diacritic restoration using text and speech representations](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 757–761, Suzhou, China. Association for Computational Linguistics.
- Ossama Obeid, Go Inoue, and Nizar Habash. 2022. [Camelira: An Arabic multi-dialect morphological disambiguator](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–326, Abu Dhabi, UAE. Association for Computational Linguistics.
- Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. [MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Sara Shatnawi, Sawsan Alqahtani, and Hanan Aldarmaki. 2024. [Automatic restoration of diacritics for speech data sets](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4166–4176, Mexico City, Mexico. Association for Computational Linguistics.
- Dimitra Vergyri and Katrin Kirchhoff. 2004. [Automatic diacritization of Arabic for acoustic modeling in speech recognition](#). In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73, Geneva, Switzerland. COLING.

Fine-Tashkeel at KSAA-2026: A Comprehensive Evaluation of Seq2Seq and Multimodal Approaches for Automatic Diacritization of Arabic Speech Dictation

Hassan Barmandah^{1,2}, Fatimah Emad Eldin^{1,3}, Omer Nacar^{1,4}, Wareef Alzubaidi^{1,5}

¹ NAMAA Community

² Umm Al-Qura University ³ Trouve Labs ⁴ Tuwaiq Academy ⁵ King Abdulaziz University

s445001043@uqu.edu.sa, Fatimah@trouve.works,

o.najar@tuwaiq.edu.sa, walzubaidi0014@stu.kau.edu.sa

Abstract

This paper presents the Fine-Tashkeel system for Task 2 of the KSAA-2026 Shared Task on Automatic Diacritization of Speech Dictation. Diacritization of speech-derived Arabic text poses challenges due to dialectal variation, morphological ambiguity, and the absence of acoustic cues in text-only pipelines. Our approach treats diacritization as a character-level sequence-to-sequence task, mapping undiacritized text directly to its fully diacritized form. We evaluate 18 models spanning text-only, ASR-augmented, and fine-tuned configurations, finding that text-only Seq2Seq approaches outperform off-the-shelf multimodal models—a gap we attribute to task mismatch in generic ASR systems rather than an inherent audio limitation. Our best submission, using zero-shot inference without task-specific training, achieved a Diacritic Error Rate (DER) of 10.56%, Word Error Rate (WER) of 34.47%, and Sentence Error Rate (SER) of 79.88%, ranking 5th out of 7 teams. Per-nationality error analysis reveals significant dialectal variation (Egyptian 3.70% vs. Algerian 13.73% DER), and diagnostic analysis confirms that case endings and vowel ambiguity are the primary bottlenecks. Code and evaluation scripts are publicly available.

Keywords: Arabic Diacritization, Sequence-to-Sequence, Speech Dictation, KSAA-2026, Tashkeel, Multimodal

1. Introduction

The KSAA-2026 Shared Task introduces a multimodal benchmark for transforming raw Arabic speech transcripts into fully diacritized text (Al Wazrah et al., 2026). Diacritic restoration remains a core challenge in Arabic NLP: a single undiacritized word (*rasm*) can have multiple valid readings depending on morphosyntactic context. Speech dictation compounds this: ASR systems produce undiacritized text, while standard diacritization models cannot leverage acoustic or dialectal cues.

We propose **Fine-Tashkeel**, a Seq2Seq system (Al-Rfooh et al., 2023) that maps undiacritized character sequences to fully diacritized counterparts. Our key outcomes:

- **Ranking:** 5th out of 7 teams (DER 10.56%, WER 34.47%).
- **Systematic Comparison:** We evaluate 18 models across text-only, ASR+Text, and fine-tuned configurations; text-only Seq2Seq consistently outperforms off-the-shelf multimodal systems due to task mismatch, not an inherent limitation of audio.
- **Error Analysis:** Per-nationality reveals dialect-dependent variation: Egyptian Arabic benefits from representation, while Gulf and Maghrebi show higher rates.

Our code and evaluation scripts are publicly available.¹

2. Background and Related Work

Arabic diacritization has evolved from rule-based systems to neural architectures. Transformer-based models (Vaswani et al., 2017) enabled sequence classification approaches by Fadel et al. (2019) and Barqawi and Zerrouki (2017). Speech-aware diacritization adds complexity: transcripts lack structural punctuation, contain dialectal variations defying MSA morphological rules, and multilingual models like mT5 (Xue et al., 2021) have shown promise for character-level generation.

Al-Rfooh et al. (2023) proposed a byte-level Seq2Seq approach that generates the full diacritized string end-to-end. For speech-aware diacritization, Shatnawi et al. (2024) demonstrated that leveraging Whisper ASR outputs as auxiliary input to a Transformer diacritization model improves performance over text-only baselines on speech data, establishing the framework adopted as the official baseline for this shared task.

3. Task Setup and Dataset

Systems must process speech audio alongside undiacritized transcripts to generate fully diacritized

¹<https://github.com/HasanBGit/KSAA2026-Fine-Tashkeel>

text (Al Wazrah et al., 2026). Hosted on Cod-aBench, the task prohibits (a) external data beyond the official training set and (b) large language models; only small-scale models are permitted. The target language covers Modern Standard Arabic (MSA) and multiple regional dialects.

3.1. Dataset Statistics

The dataset, collected via the King Salman Global Academy’s VoiceWall platform (Al Wazrah et al., 2026), comprises approximately five hours of Arabic speech. Table 1 summarizes the splits and their key characteristics.

Split	Samples	Duration	Nationalities
Train	2,327	~4.5h	9
Dev	260	~0.5h	9
Test	328	~0.7h	9

Table 1: Dataset statistics across splits.

The test set spans nine nationalities: Egypt (173), Saudi Arabia (85), Qatar (34), Sudan (10), Gaza (13), Bahrain (8), Kuwait (2), Syria (1), and Palestine (2). Algeria appears in dev but not test; Gaza in test but not dev—coverage shifts compound dialectal generalization across phonological and morphological systems.

4. System Overview

The pipeline, illustrated in Figure 1, treats diacritization as direct translation from unvoveled to fully voveled text.

4.1. Model Architecture

Our system is built on the `Fine-Tashkeel` model (Al-Rfooh et al., 2023), a Seq2Seq Transformer that, instead of classifying diacritics over a frozen string, frames diacritization as a translation task:

$$Y = \text{Seq2Seq}(X) \quad (1)$$

where $X = \{x_1, x_2, \dots, x_n\}$ is the sequence of un-diacritized characters, and $Y = \{y_1, y_2, \dots, y_m\}$ is the generated sequence containing both base characters and their corresponding diacritic marks.

This approach handles morphological fusions and can correct transcript normalization errors.

4.2. Inference Pipeline

We employ greedy search optimized for character-level precision (Appendix 10) to prioritize the most probable morphological sequence and prevent hallucinated characters, with a maximum sequence length of 1024 tokens.

5. Experimental Setup

5.1. Models Evaluated

We conducted a systematic evaluation of 18 models across four categories:

- Text-only Seq2Seq:** Fine-Tashkeel (Al-Rfooh et al., 2023) (our primary system), Shakkelha (Fadel et al., 2019), Shakkala (Barqawi and Zerrouki, 2017), mT5-base (Xue et al., 2021)
- Text-only Classifiers:** ByT5 (glonor, 2024), FLAN-T5 (Chung et al., 2024), Qwen-1.5 (Bisher, 2025), CATT (Alasmary et al., 2024), Mishkal (Zerrouki, 2020), CAMEL-MLE (Obeid et al., 2020)
- ASR+Text Multimodal:** Seamless M4T (Seamless Communication et al., 2023), ArTST (Toyin et al., 2023), Whisper variants (Radford et al., 2023; tarteel-ai, 2022; MadoggProduction, 2026)
- Fine-tuned Models:** Tashkeel-700M (Etherll, 2025), ByT5 fine-tuned (glonor, 2024), Whisper-Tashkeel fine-tuned (WajeehAzeemX, 2024)

The full list of all 18 models with their blind test scores is provided in Table 5 (Appendix 2). In addition to inference-based evaluation, we explored several training strategies—including Fine-Tashkeel fine-tuning, curriculum learning, LoRA adaptation, ensemble voting, and post-processing rules—detailed in Appendix 18. None surpassed the zero-shot Fine-Tashkeel baseline, motivating our final submission choice.

5.2. Evaluation Metrics

System performance is evaluated using three complementary metrics:

- Diacritic Error Rate (DER):** The proportion of incorrectly predicted diacritics at the character level, computed as $(S_c + D_c + I_c)/N_c$, where S_c , D_c , and I_c denote substitutions, deletions, and insertions respectively.
- Word Error Rate (WER):** The primary metric. A word is marked incorrect if it contains *any* diacritic error, making it highly sensitive to case endings (*i’rab*).
- Sentence Error Rate (SER):** A strict metric where a sentence is incorrect if any diacritic within it is wrong.

All metrics are computed under four settings by toggling two conditions: with/without case ending

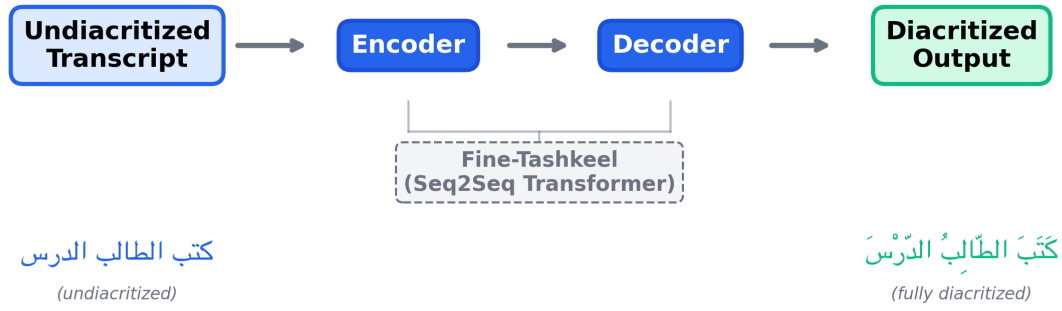


Figure 1: Overview of the Fine-Tashkeel Seq2Seq pipeline: undiacritized text is fed to the encoder; the decoder generates the fully diacritized sequence character by character.

(i'rab), and including/excluding positions with no diacritic (detailed four-setting results in Appendix 17). The official leaderboard uses the “Including no diacritic, With case ending” (WCE) setting.

6. Results

6.1. Official Leaderboard

Table 2 presents the official test leaderboard. Our Fine-Tashkeel system ranks 5th out of 7 participating teams with a DER of 10.56% and WER of 34.47%, 3.69 percentage points behind the top-ranked team. Notably, despite using only zero-shot inference without any task-specific training, our system outperforms teams 6 and 7 by a wide margin.

#	Team	DER↓	WER↓	SER↓
1	meshal	6.87	23.26	66.16
2	nadaadelmousa	7.04	24.39	71.65
3	naif_alharthi	7.51	25.34	73.48
4	nahian_abu	8.23	30.37	80.79
5	Hassan (Ours)	10.56	34.47	79.88
6	omarnj10	27.94	44.05	98.78
7	astral_fate	31.67	84.50	99.70

Table 2: Official test leaderboard (WCE, Including no diacritic). All metrics are percentages (%); lower is better. Our system (Fine-Tashkeel) ranks 5th.

6.2. Internal Model Comparison

Table 3 presents the top-performing models from our 18-configuration evaluation on the blind test set (Appendix 15). Fine-Tashkeel achieves the best DER (10.56%), followed closely by our ASR-

Tashkeel pipeline (10.70%). Development set results are reported in Appendix 4.

Model	Method	DER	WER	SER
Fine-Tashkeel	S2S / I	10.56	34.47	79.88
ASR-Tashkeel	S2S / P	10.70	—	—
Shakkelha	T / I	13.14	39.47	83.84
Shakkala	T / I	19.70	56.37	96.95
CATT	T / I	28.34	48.17	100.0
Tashkeel-700M	T / FT	34.04	57.88	99.09

Table 3: Top 6 configurations on the blind test set, ranked by DER (%). S2S = Seq2Seq, T = text-only, I = zero-shot, FT = fine-tuned, P = ASR pipeline. Full results in Appendix 2.

6.3. Text-only vs. Multimodal Comparison

A key finding is the stark contrast between text-only and multimodal (ASR+Text) models. Table 4 summarizes this comparison (Appendix 14).

Category	Avg. DER	Best DER
Text-only (Inference)	44.15	10.56
ASR+Text (Inference)	70.57	42.82
Text-only (Fine-tuned)	44.10	34.04
ASR+Text (Fine-tuned)	54.61	54.61

Table 4: Average and best blind test DER (%) by model category. Text-only models outperform ASR+Text approaches across both settings.

Despite the task involving speech dictation, multimodal models such as Seamless M4T (DER 43.01%), Whisper variants (DER 84.71–97.36%), and ArTST (DER 42.82%) underperformed significantly. **Importantly, this result should be**

interpreted with care: these off-the-shelf ASR models are optimized for transcription rather than character-level diacritic generation, and their decoders default to unvoveled output distributions. The underperformance reflects a *task mismatch*, not an inherent limitation of audio information—purpose-built multimodal architectures with task-specific acoustic adaptation could benefit from acoustic cues to disambiguate homographs.

6.4. Comparison with Organizer Baselines

Table 6 (Appendix 3) presents the organizer’s detailed baseline results on the test set across all four evaluation settings. Our inference-only DER of 10.56% (WCE, Incl. 0) is competitive with the organizer’s fine-tuned Text+ASR baseline (9.91%), despite requiring no task-specific training.

6.5. Error Analysis

A per-nationality breakdown on the development set (Table 8, Appendix 5; visualized in Appendix 11) reveals significant dialect-dependent variation. Egyptian Arabic achieves the lowest DER (3.70%), benefiting from dominant training representation (1,026/2,327 samples), while Gulf dialects (Qatar 8.74%, Saudi 10.49%, Bahrain 11.17%) and Maghrebi dialects (Algeria 13.73%) show substantially higher error rates. Per-sentence analysis (Appendix 6) shows 33.1% of sentences achieve perfect diacritization, while only 3.1% exceed 30% DER—predominantly dialectal utterances (examples in Appendix 7). A diagnostic analysis of the baseline model (Appendix 8) reveals systematic errors in sukūn, damma, and tanwīn prediction, confirming that case endings and vowel ambiguity are the primary bottlenecks.

7. Discussion

Our inference-only DER of 10.56% is competitive with the organizer’s fine-tuned baseline (9.91%), validating the Seq2Seq paradigm without task-specific training. A dev-test divergence analysis (Appendix 13) confirms Fine-Tashkeel as the most stable model (dev-to-test gap of only 2.46 pp). We acknowledge that our final submission relies on a text-only pipeline, bypassing the acoustic modality the shared task was designed to evaluate. Fine-tuning attempts encountered convergence issues (Appendix 18); off-the-shelf multimodal models proved poorly suited for character-level diacritization. As demonstrated by [Shatnawi et al. \(2024\)](#), effective speech-aware diacritization requires purpose-built architectures jointly modeling acoustic and textual features.

The 3.69 pp gap from the top-ranked system suggests avenues for improvement: (1) task-specific fine-tuning with larger corpora; (2) purpose-built multimodal architectures fusing acoustic embeddings with text representations; and (3) data augmentation targeting underrepresented dialects (Appendix 5).

8. Conclusion

We presented Fine-Tashkeel for the KSAA-2026 Shared Task, ranking 5th of 7 teams (DER 10.56%, WER 34.47%). Through evaluation of 18 models, we demonstrate that: (1) character-level Seq2Seq models outperform off-the-shelf multimodal and classifier-based approaches, reflecting task mismatch rather than an inherent limitation of acoustic information; (2) text-only inference achieves performance competitive with fine-tuned baselines; and (3) dialectal variation is the dominant error source (Egyptian 3.70% vs. Algerian 13.73% DER).

Limitations

Our system has limitations. First, as a text-only approach, it cannot leverage acoustic cues that could disambiguate homographs. Second, performance degrades on underrepresented dialects (Algerian, Bahraini). Third, we did not successfully fine-tune the Fine-Tashkeel model on the shared task data due to convergence issues, relying instead on zero-shot inference. Fourth, our evaluation is limited to the relatively small VoiceWall dataset (~5 hours), and generalization to other domains or larger corpora remains untested.

Ethics Statement

This work supports the development of Arabic language technology, specifically diacritization of speech dictation data. The shared task data were collected via the King Salman Global Academy’s VoiceWall platform; consent and curation are the task organizers’ responsibility, and we use the data as provided. Automated diacritization may contain errors; we recommend human validation before use in educational, religious, or accessibility applications. Our code supports reproducibility.

Acknowledgements

We thank the King Salman Global Academy for Arabic Language for providing the VoiceWall dataset and organizing the shared task ([Al Wazrah et al., 2026](#)). We also acknowledge the baseline implementations provided by [Shatnawi et al. \(2024\)](#).

References

- Bashar Al-Rfooh, Gheith Abandah, and Rami Al-Rfou. 2023. [Fine-tashkeel: Finetuning byte-level models for accurate arabic text diacritization](#). ArXiv preprint arXiv:2303.14588.
- Asma Al Wazrah, Waad Alshammari, Rawan Almatham, Raghad Al-Rasheed, Afrah Altamimi, Rufael Marew, Sawsan Alqahtani, Hanan Aldarmaki, Abdullah Alharbi, Abdulrahman Alshehri, Mohamed Assar, Amal Almazrui, and Abdulrahman AlOsaimey. 2026. KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*.
- Faris Alasmari, Orjuwan Zaafarani, and Ahmad Ghannam. 2024. [CATT: Character-based Arabic tashkeel transformer](#). In *Proceedings of the Second Arabic Natural Language Processing Conference (ArabicNLP 2024)*, pages 250–257, Bangkok, Thailand. Association for Computational Linguistics.
- Ahmad Barqawi and Taha Zerrouki. 2017. Shakkala: Arabic text vocalization. <https://github.com/Barqawiz/Shakkala>.
- Bisher. 2025. qwen-1p5-diacritization. <https://huggingface.co/Bisher/qwen-1p5-diacritization>. Accessed: March 2026.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25:1–53.
- Etherll. 2025. Tashkeel-700m model. <https://huggingface.co/Etherll/Tashkeel-700M>. Accessed: March 2026.
- Ali Fadel, Ibraheem Tuffaha, Bara’ Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019. [Neural Arabic text diacritization: State of the art results and a novel approach for machine translation](#). In *Proceedings of the 6th Workshop on Asian Translation*, pages 215–225, Hong Kong, China. Association for Computational Linguistics.
- glonor. 2024. byt5-arabic-diacritization. <https://huggingface.co/glonor/byt5-arabic-diacritization>. Accessed: March 2026.
- MaddoggProduction. 2026. whisper-l-v3-turbo-quran-lora-dataset-mix. <https://huggingface.co/MaddoggProduction/whisper-l-v3-turbo-quran-lora-dataset-mix>. Accessed: March 2026.
- Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Alexander Erdmann, and Nizar Habash. 2020. CAMEL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032. European Language Resources Association.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of PMLR, pages 28492–28518.
- Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Mailard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinash Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta R. Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Hefernan, Somya Jain, Justine Kao, Ann Lee, Xutai Ma, Alex Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Sun, Paden Tomasello, Changan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. 2023. Seamless: Multilingual expressive and streaming speech translation. In *ArXiv preprint arXiv:2312.05187*.
- Sara Shatnawi, Sawsan Alqahtani, and Hanan Aldarmaki. 2024. Automatic restoration of diacritics for speech data sets. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4166–4176. Association for Computational Linguistics.
- tarteel-ai. 2022. whisper-base-ar-quran. <https://huggingface.co/tarteel->

[ai/whisper-base-ar-quran](#). Accessed: March 2026.

Hawau Toyin, Amirbek Djanibekov, Ajinkya Kulkarni, and Hanan Aldarmaki. 2023. [ArTST: Arabic text and speech transformer](#). In *Proceedings of ArabicNLP 2023*, pages 41–51, Singapore (Hybrid). Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 6000–6010.

WajeehAzeemX. 2024. [whisper-tiny-ar-tashkeel](#). <https://huggingface.co/WajeehAzeemX/whisper-tiny-ar-tashkeel>. Accessed: March 2026.

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. Association for Computational Linguistics.

Taha Zerrouki. 2020. *Towards An Open Platform For Arabic Language Processing*. Phd thesis, Ecole Nationale Supérieure d’informatique, Alger, Algérie.

1. Evaluation Metrics

System performance is evaluated using three complementary metrics. Let N_c , N_w , and N_s represent the total number of characters, words, and sentences in the reference text, respectively.

1. Diacritic Error Rate (DER): Measures the proportion of incorrectly predicted diacritics.

$$\text{DER} = \frac{S_c + D_c + I_c}{N_c} \quad (2)$$

2. Word Error Rate (WER): The primary metric. A word is marked incorrect if it contains at least one diacritic error, making it highly sensitive to case endings (*i’rab*).

$$\text{WER} = \frac{\text{Incorrect Words}}{N_w} \quad (3)$$

3. Sentence Error Rate (SER): A strict metric where a sentence is incorrect if any diacritic within it is flawed.

$$\text{SER} = \frac{\text{Incorrect Sentences}}{N_s} \quad (4)$$

These metrics are computed under four settings by toggling two binary conditions: (a) whether case endings (*i’rab*) are considered, and (b) whether character positions with no diacritic are included in the count. The official leaderboard uses “With case ending, Including no diacritic” (WCE, Incl. 0).

2. Full Internal Results

3. Organizer Baseline Results

4. Development Set Analysis

Fine-Tashkeel achieves the best dev DER (8.10%) among all models with reliable dev evaluation, followed by Shakkelha (10.26%) and Shakkala (11.64%). Fine-Tashkeel also shows the most consistent dev-to-test behavior, with a modest gap of only 2.46 percentage points (8.10% → 10.56%), indicating robust generalization.

5. Per-Nationality Detailed Analysis

Table 8 presents the full per-nationality breakdown of DER, WER, and SER on the development set, with and without case ending evaluation.

5.1. Gender Analysis

The model shows a modest gender gap: male speakers achieve 7.78% DER (WCE) compared to 8.62% for female speakers, a difference of 0.84 percentage points. This is likely due to the higher proportion of Egyptian male speakers in the dataset rather than an inherent gender bias in the model.

6. Error Distribution Analysis

Table 9 shows the distribution of per-sentence DER on the development set.

The average case ending impact on DER is −5.51%, meaning that removing case ending evaluation reduces DER by approximately 5.5 percentage points on average. This indicates that *i’rab* (grammatical case marking) remains a significant source of error, particularly for informal and dialectal speech where case endings are often phonologically reduced or absent.

7. Qualitative Error Examples

Table 10 presents representative error cases from the worst-performing predictions on the dev set. The errors fall into three main categories:

Model / Submission Name	Type	Method	DER (Dev)	DER (Blind) ↓	WER (Blind) ↓	SER (Blind) ↓
Inference (no task-specific training)						
Fine-Tashkeel (Al-Rfooh et al., 2023)	Seq2Seq	Inference	8.10	10.56	34.47	79.88
Shakkelha (Fadel et al., 2019)	Text-only	Inference	10.26	13.14	39.47	83.84
Shakkala (Barqawi and Zerrouki, 2017)	Text-only	Inference	11.64	19.70	56.37	96.95
CATT (Alasmary et al., 2024)	Text-only	Inference	—	28.34	48.17	100.00
FLAN-T5 (Chung et al., 2024)	Text-only	Inference	24.37	37.80	59.64	100.00
ArTST (Toyin et al., 2023)	ASR+Text	Inference	—	42.82	65.94	85.06
Seamless M4T (Seamless Communication et al., 2023)	ASR+Text	Inference	—	43.01	56.57	100.00
CAMEL-MLE (Obeid et al., 2020)	Text-only	Inference	28.71	45.89	82.39	100.00
ByT5 Glonor (glonor, 2024)	Text-only	Inference	15.28	46.96	69.64	100.00
Qwen-1.5 (Bisher, 2025)	Text-only	Inference	33.75	62.56	75.75	97.87
Mishkal (Zerrouki, 2020)	Text-only	Inference	17.37	76.76	90.49	99.39
Whisper (large-v3) (Radford et al., 2023)	ASR+Text	Inference	—	84.71	99.60	100.00
Tarteel Whisper (tarteel-ai, 2022)	ASR+Text	Inference	—	84.93	85.08	100.00
Whisper Quran LoRA (MaddoggProduction, 2026)	ASR+Text	Inference	—	97.36	99.98	100.00
mT5-base (Xue et al., 2021)	Text-only	Inference	82.17	99.74	99.97	100.00
Fine-tuned on task data						
Tashkeel-700M (Etherll, 2025)	Text-only	Fine-tune	22.08	34.04	57.88	99.09
Whisper Tiny Ar Tashkeel (WajeelAzeemX, 2024)	ASR+Text	Fine-tune	46.05	54.61	67.68	99.39
ByT5 Glonor (glonor, 2024)	Text-only	Fine-tune	23.12	54.15	77.57	100.00

Table 5: Comprehensive results for all 18 models evaluated. Models are grouped by method (inference vs. fine-tune) and sorted by blind test DER within each group. Dev scores computed using our evaluation script; blind test scores from the competition platform. “—” indicates dev scores excluded due to an evaluation library artifact that post-processed predictions. All values are percentages (%).

Evaluation Setting (%)	Text+ASR			Text-only			Fine-Tuned Text+ASR		
	DER	WER	SER	DER	WER	SER	DER	WER	SER
Including no diacritic									
With case ending	13.50	40.24	82.32	17.66	49.85	91.77	9.91	31.84	82.93
Without case ending	10.58	27.95	71.95	13.23	32.24	82.62	7.89	20.99	67.07
Excluding no diacritic									
With case ending	14.26	33.03	75.61	20.08	46.20	91.77	8.52	24.73	78.66
Without case ending	9.96	19.71	60.37	13.93	27.07	81.71	4.82	10.89	50.61

Table 6: Detailed organizer baseline results on the **test set**. Columns show DER/WER/SER for each configuration. Our inference-only DER (10.56%, WCE, Incl. 0) is competitive with the organizer’s fine-tuned Text+ASR baseline (9.91%).

- Dialectal Mismatch:** The model applies MSA diacritization patterns to dialectal forms (e.g., Gazan عاوزه ‘*āwzah*’ ‘she wants’ receives MSA vowelizing).
- Morphological Ambiguity:** Words with multiple valid readings receive the wrong diacritic pattern (e.g., مجريات ‘*mujariyat*’ vs. ‘*mujriyat*’).
- Transliteration Errors:** Foreign terms and proper nouns are incorrectly diacritized due to absence from training data.

8. Baseline Diagnostic Analysis

To better understand the challenges of diacritizing speech-derived text, we conduct a diagnostic analysis using the official text-only baseline model on the development set. The baseline is a Transformer-based diacritization model pretrained

on the Tashkeela corpus, operating purely on undiacritized text without access to acoustic information. While such models perform well on written Modern Standard Arabic (MSA), we observe substantial degradation when applied to speech transcripts in the KSAA dataset.

Sentence-Level Errors Using a strict Sentence Error Rate (SER) criterion—where a sentence is considered incorrect if any diacritic differs from the reference—we find that **95.0%** of sentences (247/260) contain at least one error. This reflects the inherently strict nature of full diacritization: even a single incorrect diacritic leads to a full sentence error. Nevertheless, the magnitude of this result highlights the difficulty of the task on speech-derived inputs.

Case Ending Errors Case endings (*i’rāb*) emerge as a dominant source of error. By com-

Rank	Model	DER (WCE)	WER (WCE)	SER (WCE)	DER (w/o CE)	WER (w/o CE)
1	Fine-Tashkeel	8.10	20.46	67.31	13.86	43.66
2	Shakkelha	10.26	24.37	72.31	15.92	47.03
3	Shakkala	11.64	27.89	78.85	17.35	50.02
4	ByT5 Glonor	15.28	38.43	94.23	20.59	56.84

Table 7: Models with reliable dev set scores, ranked by DER (WCE, Incl. 0). Several other models (ArTST, CATT, Seamless M4T, Ensemble, Postprocess) were excluded because their dev scores were artifacts of an evaluation library that inadvertently post-processed the predictions.

Nationality	N	DER (WCE)	WER (WCE)	SER (WCE)	DER (w/o CE)	WER (w/o CE)
Egypt	114	3.70	9.01	42.98	4.88	15.62
Palestine	8	4.71	12.66	75.00	14.67	55.70
Kuwait	12	5.75	14.86	50.00	15.49	53.38
Syria	1	5.65	12.50	100.00	16.03	62.50
Qatar	24	8.74	20.45	87.50	19.63	58.47
Saudi	41	10.49	28.67	90.24	18.17	60.37
Sudan	19	10.86	30.08	89.47	17.20	57.32
Bahrain	13	11.17	34.87	92.31	18.98	61.18
Algeria	28	13.73	35.05	92.86	20.79	62.70
Overall	260	8.10	20.46	67.31	13.86	43.66

Table 8: Full per-nationality DER/WER/SER breakdown on the dev set for Fine-Tashkeel, with and without case ending evaluation. The gap between best (Egypt, 3.70% DER) and worst (Algeria, 13.73% DER) nationality is 10.03 percentage points.

DER Range	Count	Percentage
0% (Perfect)	86	33.1%
0–5%	41	15.8%
5–10%	55	21.1%
10–15%	32	12.3%
15–20%	17	6.5%
20–30%	21	8.1%
30–50%	8	3.1%
Total	260	100%

Table 9: Distribution of per-sentence DER (WCE, Incl. 0) on the dev set. One-third of sentences are perfectly diacritized; the tail (>30% DER) consists exclusively of dialectal utterances.

paring the final diacritic of each sentence, we observe that **55.4%** of sentences exhibit incorrect case endings. This confirms that syntactically governed diacritics are particularly challenging to recover from text alone, especially in conversational and dialectal speech where such endings are often reduced, omitted, or ambiguous.

Diacritic Distribution Shift We compare the distribution of predicted diacritics against the ground truth. Three systematic failure modes emerge:

- **Sukūn underprediction** (−26.7%): difficulty modeling consonant closure, leading to 605 missed zero-vowel markers—the highest ab-

solute error count.

- **Damma overprediction** (+26.3%): confusion between vowel classes in ambiguous phonological contexts, contributing 231 errors.
- **Tanwīn overprediction**: severe systematic bias across all tanwīn types. Dammatan +166.7% (40 errors), Kasratan +54.1% (46 errors), Fathatan +43.7% (31 errors), indicating difficulty distinguishing indefinite case marking from standard vowel assignments.

High-frequency diacritics show contrasting patterns: *faṭḥa* exhibits minor deviations (3.3%), while *sukūn* shows substantial errors (26.7%), indicating that errors concentrate in phonetically subtle (consonant closure) and grammatically complex (case marking) categories rather than simply low-frequency ones.

These results complement the modality comparison in Section 6: text-only models struggle with vowel ambiguity and case endings due to missing phonetic cues, while off-the-shelf multimodal systems fail to compensate due to task mismatch and error propagation (additional details in Appendix 9).

ID	DER	Reference	Prediction
Gaza_025	28.6	مَحْرَقَةُ رِفْهِ الْعَدُوِّ يَنْتَقِمُ مِنَ الْأَمِينِ <i>maḥraqatu rafaḥ al-'aduww yantaqim min al-āminīn</i> 'The massacre of Rafah; the enemy takes revenge on the safe ones'	مَحْرَقَةُ رِفْهِ الْعَدُوِّ يَنْتَقِمُ مِنَ الْأَمِينِ <i>muḥriqatu rifḥi al-'aduww yantaqimu min al-āminīna</i>
Qatar_422	32.5	إِسْلِمَ عَلَى رَفِيجِكَ تَأْخُذُ تُكْرِمُ جَارَكَ <i>itsallim 'alā rafijik tākhidh tikrim jārak</i> 'Greet your friend, take [gifts], honor your neighbor'	إِسْلِمَ عَلَى رَفِيجِكَ تَأْخُذُ تُكْرِمُ جَارَكَ <i>itsalim 'alā rafijika tākhudh tukrim jāraka</i>
Sudan_121	32.6	وَأَمْتَحَنَّا وَاجْتَرْنَا الْأَمْتَحَنَاتِ <i>wa-imtiḥannā wa-ijṭaznā al-imtiḥānāt</i> 'We were tested and we passed the exams'	وَأَمْتَحَنَّا وَاجْتَرْنَا الْأَمْتَحَنَاتِ <i>wa-mtaḥannā wa-jṭaznā al-imtiḥānāti</i>
Egypt_287	33.3	ثَنَا إِسْمَاعِيلُ بْنُ عِيَّاشٍ <i>thanā ismā'īlu bnu 'ayyāshin</i> 'Isma'il ibn Ayyash narrated to us'	ثَنَا إِسْمَاعِيلُ بْنُ عِيَّاشٍ <i>thanā ismā'īlu bnu 'ayyāshin</i>

Table 10: Representative worst-case error examples from the dev set. DER is per-sentence (%), WCE). Each cell shows the Arabic text, its transliteration, and (for reference) an English translation. Errors predominantly occur on dialectal forms and morphologically ambiguous words.

9. Text-only vs. Multimodal: Additional Details

The main text (Section 6) presents our modality comparison and its interpretation. Here we provide additional context on individual multimodal model failures.

Whisper-based models fine-tuned on Quranic Arabic (Tarteel Whisper, DER 84.93%; Whisper Quran LoRA, DER 97.36%) exhibit the most severe domain mismatch: formal tajweed diacritization rules are incompatible with conversational dialectal speech. The standard Whisper large-v3 model (DER 84.71%) produces predominantly unvoiced transcriptions, confirming that its decoder distribution is optimized for transcription rather than diacritization. Seamless M4T (DER 43.01%) and ArTST (DER 42.82%) perform better but still far below text-only approaches, as their architectures are not designed for character-level diacritic generation.

10. Sequence-to-Sequence Configuration

The *Fine-Tashkeel* model was configured with the following generation parameters to ensure deterministic and accurate character rendering:

- **Decoding Strategy:** Greedy Search (`do_sample = False`) to prioritize the most mathematically probable morphological sequence and prevent hallucinated characters.

- **Max Length:** 1024 tokens, sufficient to cover utterances in the dataset.
- **Penalty Formulation:** Length penalty applied to ensure the model does not prematurely truncate final case endings (*i'rab*).

11. Per-Nationality Visualization

Figure 2 visualizes the per-nationality DER and WER performance of *Fine-Tashkeel* on the development set. Nationalities are sorted by increasing DER and color-coded by dialect region: Nile Valley (Egypt, Sudan), Levantine (Palestine, Syria), Gulf (Kuwait, Qatar, Saudi, Bahrain), and Maghrebi (Algeria). The strong correlation between training data representation and model performance is evident, with Egypt ($n = 114$) achieving the lowest error rates and Algeria ($n = 28$) the highest.

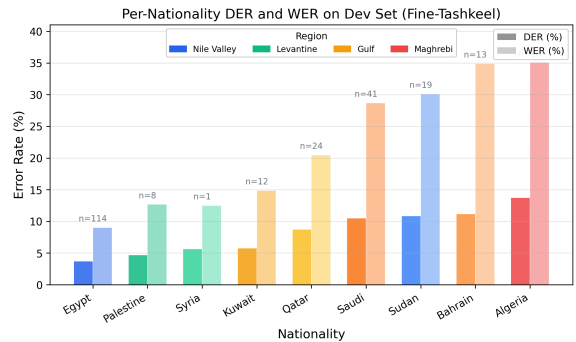


Figure 2: Per-nationality DER and WER (%) on the dev set, color-coded by dialect region. Sample counts are shown above each bar.

12. DER Distribution Visualization

Figure 3 presents the distribution of per-sentence DER values across the 260 development set samples. The distribution is heavily right-skewed: 33.1% of sentences achieve perfect diacritization (DER = 0%), while only 3.1% exceed 30% DER. This bimodal pattern suggests the model handles MSA-like and frequent dialectal patterns well, but struggles with rare dialectal forms.

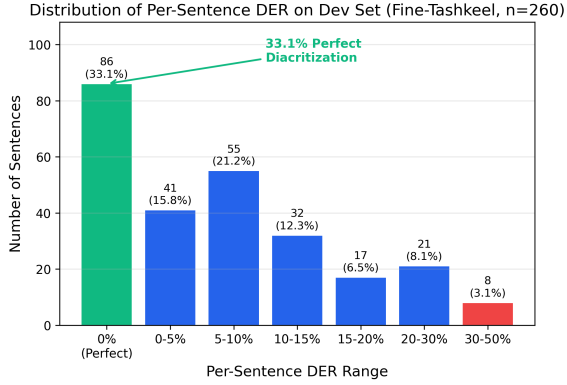


Figure 3: Distribution of per-sentence DER on the dev set ($n = 260$). One-third of sentences are perfectly diacritized; the tail consists exclusively of dialectal utterances.

13. Dev-Test Divergence Analysis

Figure 4 plots each model’s dev DER (x-axis) against its blind test DER (y-axis) for models with reliable dev evaluation. Points on the diagonal indicate consistent performance; points far above indicate overfitting or distribution shift.

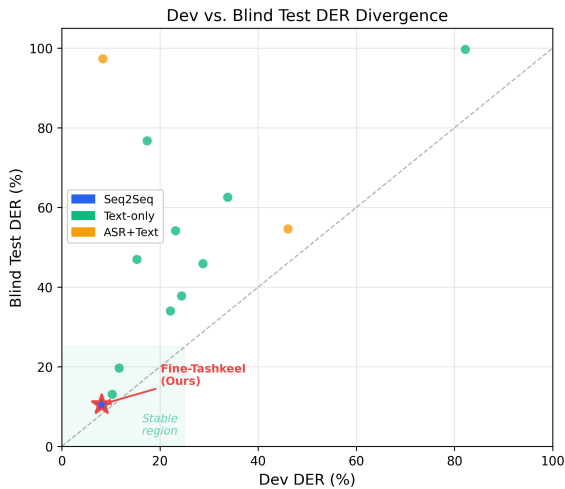


Figure 4: Dev vs. blind test DER for models with reliable dev evaluation. Fine-Tashkeel (star) shows the most stable generalization behavior.

Fine-Tashkeel is the most stable model among those with reliable dev evaluation, with a modest dev-to-test increase of only 2.46 percentage points (8.10% $\rightarrow$ 10.56%). Models with larger dev-to-test gaps (e.g., Shakkala: +8.06 pp, ByT5 Glonor: +31.68 pp) suggest sensitivity to dialectal distribution shift between the dev and test splits. Table 11 quantifies these divergences.

Model	Dev DER	Test DER	Δ
Fine-Tashkeel	8.10	10.56	+2.46
Shakkalha	10.26	13.14	+2.88
Shakkala	11.64	19.70	+8.06
ByT5 Glonor	15.28	46.96	+31.68
Mishkal	17.37	76.76	+59.39
FLAN-T5	24.37	37.80	+13.43
Qwen-1.5	33.75	62.56	+28.81

Table 11: Dev-to-test DER divergence (Δ) for models with reliable dev evaluation. Fine-Tashkeel shows the smallest gap, indicating robust generalization.

14. Text-only vs. Multimodal Visualization

Figure 5 visualizes the modality comparison presented in the main text (Table 4).

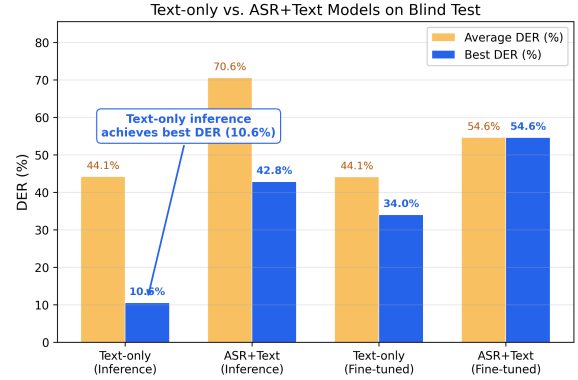


Figure 5: Average and best blind test DER by model category. Text-only inference models outperform ASR+Text approaches across all settings.

As discussed in the main text, this result reflects the task mismatch of available off-the-shelf models rather than an inherent limitation of incorporating acoustic information.

15. Full Model Comparison Visualization

Figure 6 provides a visual overview of all 18 evaluated configurations ranked by blind test DER. Models are color-coded by type (Seq2Seq, Text-only,

ASR+Text) and shading distinguishes inference from fine-tuned models.

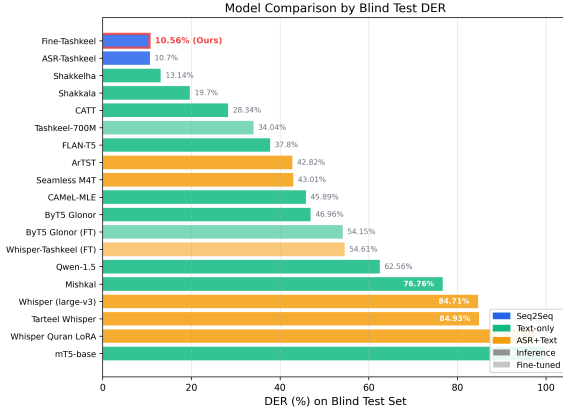


Figure 6: All evaluated models ranked by blind test DER. Fine-Tashkeel (highlighted) achieves the best DER among all configurations tested.

The visualization clearly shows the performance clustering: the top three models (Fine-Tashkeel, ASR-Tashkeel, Shakkella) all fall below 15% DER and are all text-based Seq2Seq or text-only systems. The long tail of high-DER models is dominated by ASR+Text approaches, confirming that off-the-shelf multimodal models are poorly suited for this task.

16. ASR-Tashkeel Pipeline

Our second-best model (DER 10.70%) uses a two-stage ASR-Tashkeel pipeline that combines Whisper-based transcription with Fine-Tashkeel diacritization:

- Stage 1: ASR Transcription.** The input audio is processed by Whisper (large-v3) (Radford et al., 2023) to generate an undiacritized Arabic transcript. We use the standard inference configuration with Arabic language detection.
- Stage 2: Seq2Seq Diacritization.** The ASR-generated transcript is passed to Fine-Tashkeel (Al-Rfooh et al., 2023), which adds full diacritization using the same configuration described in Appendix 10.

This pipeline approach achieves a blind test DER of 10.70%, only 0.14 percentage points behind the text-only Fine-Tashkeel model (10.56%). The near-identical performance is attributable to Whisper’s high-quality Arabic transcription: since the pipeline’s diacritization stage receives a clean transcript, the quality bottleneck shifts to the diacritization model itself rather than transcription errors.

This finding validates the use of ASR as a preprocessing stage when the ground-truth transcript is unavailable, while confirming that the clean transcript pathway remains marginally superior.

17. Detailed Four-Setting Results

Tables 12 and 13 present Fine-Tashkeel’s results across all four evaluation settings on the development and test sets respectively, providing insight into the relative contributions of case ending errors and no-diacritic positions.

Setting	DER	WER	SER
<i>Including no diacritic</i>			
With case ending	8.10	20.46	67.31
Without case ending	13.86	43.66	83.46
<i>Excluding no diacritic</i>			
With case ending	7.43	14.90	58.46
Without case ending	15.13	39.71	77.31

Table 12: Fine-Tashkeel **dev set** performance across all four evaluation settings.

Setting	DER	WER	SER
<i>Including no diacritic</i>			
With case ending	10.56	34.47	79.88
Without case ending	7.46	20.81	68.29
<i>Excluding no diacritic</i>			
With case ending	11.56	29.45	76.83
Without case ending	6.86	14.32	64.33

Table 13: Fine-Tashkeel **test set** performance across all four evaluation settings (official Co-daLab scores). Removing case endings reduces WER from 34.47% to 20.81% (−13.66 pp), confirming *i’rāb* as the primary error source.

The results reveal two important patterns on the dev set:

- Case ending effect:** Evaluating without case ending (*i’rāb*) increases DER from 8.10% to 13.86% (+5.76 pp). This counterintuitive result occurs because the “without case ending” setting excludes word-final positions from evaluation, removing positions where the model often predicts correctly (since many word-final diacritics are straightforward). The remaining interior positions have a higher proportion of ambiguous diacritics.
- No-diacritic effect:** Excluding positions with no diacritic (Excl. 0) reduces DER from 8.10% to 7.43% (−0.67 pp), indicating the model

handles consonant clusters and sukun predictions well. The most challenging positions are those requiring active diacritic selection.

On the test set (Table 13), the pattern is more intuitive: removing case endings *reduces* WER from 34.47% to 20.81% (−13.66 pp), confirming that case endings are the primary error source on unseen data. Under the most lenient setting (WOCE, Excl. 0), our system achieves 6.86% DER and 14.32% WER. The full four-setting comparison across all teams is provided in Table 15 (Appendix 19).

Table 14 compares the top 4 models across both primary settings on the development set.

Model	DER (WCE, Incl. 0)	WER (WCE, Incl. 0)	DER (w/o CE, Incl. 0)	WER (w/o CE, Incl. 0)
Fine-Tashkeel	8.10	20.46	13.86	43.66
Shakkelha	10.26	24.37	15.92	47.03
Shakkala	11.64	27.89	17.35	50.02
ByT5 Glonor	15.28	38.43	20.59	56.84

Table 14: Top 4 models with reliable dev scores across two evaluation settings (WCE and without case ending, both Incl. 0).

18. Training Strategy Exploration

In addition to our primary zero-shot inference approach, we explored several training and adaptation strategies during development. None surpassed the zero-shot Fine-Tashkeel baseline on the blind test set, but they provide useful context for our final system choice.

18.1. Fine-Tashkeel Fine-Tuning

We attempted to fine-tune the Fine-Tashkeel model on the shared task training data (2,327 samples) for 5 epochs with a learning rate of 2×10^{-5} , batch size 30, and cosine learning rate scheduler. Data augmentation was applied via word dropout and character noise injection (yielding 3,523 augmented samples). However, the model failed to converge, producing 100% DER on the test set. This is discussed further in Limitations.

18.2. Curriculum Learning (ByT5)

We implemented a 3-stage progressive curriculum training strategy using the ByT5-Glonor model (glonor, 2024): (1) training on short, high-frequency MSA phrases; (2) adding medium-length dialectal utterances; (3) full dataset training. This achieved a dev DER of 28.19%, underperforming the standard ByT5 fine-tuning (dev

DER 23.12%) and far below the zero-shot Fine-Tashkeel baseline (8.10%).

18.3. LoRA Fine-Tuning (Tashkeel-700M)

We applied Low-Rank Adaptation (LoRA) to the Tashkeel-700M model (Etherll, 2025) with rank $r = 16$, $\alpha = 32$, and 4-bit quantization to reduce memory requirements. This yielded a dev DER of 27.59%, comparable to the curriculum learning approach but similarly unable to match the zero-shot baseline.

18.4. ASR-Tashkeel Pipeline

Our two-stage Whisper+Fine-Tashkeel pipeline (detailed in Appendix 16) achieved a blind test DER of 10.70%, our second-best result and only 0.14 pp behind the text-only approach.

18.5. Ensemble Voting

We implemented majority voting across 6 models (Fine-Tashkeel, Shakkelha, Shakkala, ByT5, ArTST, CATT) at the character level. The ensemble achieved a dev DER of 7.48%, slightly outperforming the best individual model on the dev set (Fine-Tashkeel, 8.10%), but showing no improvement on the blind test set. This suggests the models share correlated errors on difficult dialectal examples.

18.6. Post-Processing Rules

We implemented rule-based post-processing including shadda reordering (ensuring shadda precedes the vowel diacritic) and tanween validation (checking consonant compatibility). These rules had minimal to slightly negative impact on DER, indicating that the Fine-Tashkeel model already produces well-formed diacritic sequences.

19. Official Leaderboard: Four-Setting Results

Table 15 presents the complete official test results for all 7 participating teams across all four evaluation settings, as reported on the CodaLab leaderboard. This extends the main leaderboard (Table 2), which reports only the primary setting (WCE, Incl. 0).

A key insight from this table is the impact of case endings on our system’s performance. When excluding case endings (WOCE), our WER drops from 34.47% to 20.81%—a reduction of 13.66 percentage points—confirming that *i’rab* is the dominant error source. Furthermore, under the most lenient setting (WOCE, Excl. 0), our WER of 14.32%

Team	WCE, Incl. 0			WOCE, Incl. 0			WCE, Excl. 0			WOCE, Excl. 0		
	DER	WER	SER	DER	WER	SER	DER	WER	SER	DER	WER	SER
meshal	6.87	23.26	66.16	5.97	17.05	53.96	6.64	18.64	64.33	4.80	10.48	50.91
nadaadelmousa	7.04	24.39	71.65	6.17	17.96	55.18	6.92	18.97	68.90	5.15	11.06	50.61
naif_alharthi	7.51	25.34	73.48	6.60	18.66	60.06	7.51	19.95	70.73	5.68	11.85	55.49
nahian_abu	8.23	30.37	80.79	6.76	20.84	64.02	8.46	24.33	74.39	5.87	13.43	55.18
Hassan (Ours)	10.56	34.47	79.88	7.46	20.81	68.29	11.56	29.45	76.83	6.86	14.32	64.33
omarnj10	27.94	44.05	98.78	27.68	36.77	97.87	25.38	38.43	98.48	25.27	31.07	97.56
astral_fate	31.67	84.50	99.70	19.86	50.45	99.39	41.30	86.33	99.70	25.03	49.07	99.39

Table 15: Official test leaderboard across all four evaluation settings. All values are percentages (%); lower is better. WCE/WOCE = with/without case ending; Incl./Excl. 0 = including/excluding positions with no diacritic. Our system’s WER drops from 34.47% (WCE) to 20.81% (WOCE), indicating case endings are the primary error source.

is competitive with nahian_abu (13.43%), narrowing the gap to just 0.89 pp. This suggests that our system’s core diacritization capability is strong, and targeted improvements in case ending prediction could yield substantial gains in the primary metric.

Eraserhead at OSACT7 Shared Task: ASR Consistency Filtering and Speaker-Adaptive Post-Processing for Arabic Speech Diacritization

Muhammad Abu Horaira, Nahian Chowdhury

Chittagong University of Engineering and Technology

Chittagong, Bangladesh

{u2004029,u2004026}@student.cuet.ac.bd

Abstract

Arabic speech diacritization is the task of restoring short vowel marks to undiacritized text derived from speech input. It remains difficult because ASR output can be noisy, dialectal variation is substantial, and speakers often differ in how they realize word-final diacritics. In this paper, we describe our submission to Task 2 of the KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization, where our system ranked 4th on the official leaderboard. Our approach builds on a pretrained ASR-aware diacritization model and adds three components: ASR Consistency Filtering, confidence-based ensembling of three checkpoints, and speaker-adaptive post-processing specifically for word-final diacritics. Rather than discarding problematic data, our filtering strategy replaces unreliable ASR transcripts with the undiacritized gold text rather than removing training examples, which makes training more stable. On the official test set, our system achieved a Diacritic Error Rate (DER) of 8.23, a Word Error Rate (WER) of 30.37, and a Sentence Error Rate (SER) of 80.79 under the With Case Endings (WCE), Including No Diacritic (Incl. 0) evaluation setting. It also outperformed the organizers' fine-tuned Text+ASR baseline in three of the four main evaluation settings.

Keywords: Arabic diacritization, automatic speech recognition, speech-aware diacritization, consistency filtering

1. Introduction

Arabic diacritization, the task of restoring of short vowels and symbols to undiacritized text, remains a significant challenge in Arabic NLP. Their absence in most writing creates lexical and syntactic ambiguity for both humans and systems (Elgamel et al., 2024). Hierarchical neural approaches operating at both word and character levels have shown strong results on benchmark text diacritization datasets (AlKhamissi et al., 2020). However, text-based diacritization models are less effective on speech transcripts, where spoken-language variation leads to substantially higher error rates (Shatnawi et al., 2024). This challenge becomes even greater in ASR output, which may contain recognition errors, spurious tokens, and dialectal variation that diverges from the assumptions of text-only diacritizers.

The KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization (Al Wazrah et al., 2026) (Task 2) targets this problem directly. Given a speech utterance and its undiacritized transcript, systems must produce a fully diacritized output at the character level, predicting all diacritic marks, including fatha, damma, kasra, sukūn, shaddah, and tanwīn. The dataset covers Modern Standard Arabic and dialectal speech from multiple speaker backgrounds, introducing additional variation in diacritic realization.

We present a system built on the Diac Trans-former (Shatnawi et al., 2024) and make the following contributions:

- We introduce *ASR Consistency Filtering*, which replaces unreliable ASR transcripts with the undiacritized gold text based on character-level overlap, rather than discarding training examples.
- We train three checkpoints, two with different filtering thresholds ($\tau=0.5$ and $\tau=0.9$) and one with $\tau=0.5$ and a different random seed, and combine them using character-level confidence-voting ensemble decoding.
- We apply speaker-adaptive post-processing that normalizes word-final diacritics based on per-speaker stylistic patterns estimated from development-set annotations.

Our system achieved 8.23 DER and 30.37 WER on the primary test setting, ranking 4th on the official leaderboard and outperforming the organizers' fine-tuned Text+ASR baseline in three of four evaluation conditions.

2. Related Work

Arabic diacritization has been widely studied using neural sequence models. Fadel et al. (2019) reported strong results on the Tashkeela benchmark, while Mubarak et al. (2019) showed that sequence-to-sequence models with voting can outperform earlier feature-based approaches. Later, Qin et al. (2021) showed that even noisy auxiliary information, such as automatically generated diacritizations, can still be useful when combined with regularized decoding and adversarial training. Similarly,

Thompson and Alshehri (2022) showed that multi-task learning with translation as an auxiliary task can improve Arabic diacritization by injecting additional semantic and linguistic information. These results suggest that external or imperfect auxiliary signals can still be beneficial when handled carefully, which is closely related to the motivation behind our ASR consistency filtering.

Recent work has also explored the use of speech-related information for Arabic diacritization. Shatnawi et al. (2024) used diacritized ASR transcripts as a second input through cross-attention and reported substantial reductions in diacritic error rates on speech data. Aldarmaki and Ghannam (2023) also showed that ASR-aware diacritization can outperform applying text-only diacritization as a post-processing step. Building on this line of work, our system adds consistency-based filtering for noisy ASR inputs and speaker-adaptive post-processing for word-final diacritics.

3. Dataset

We used the official KSAA-2026 shared-task dataset. The corpus contains approximately five hours of Arabic speech covering Modern Standard Arabic and dialectal speech from nine nationalities. The official splits consist of 2,327 training utterances (≈ 4.5 h), 260 development utterances (≈ 0.5 h), and 260 test utterances (≈ 0.5 h). All recordings are 16 kHz mono WAV files of at most nine seconds. Task 2 is defined as speech dictation with automatic diacritization, where systems take speech audio together with undiacritized text and produce a fully diacritized transcript. In our experiments, the diacritized references provided in the official data were used as supervision, and undiacritized text was obtained by removing diacritic marks when needed. The cross-dialect and cross-speaker diversity motivated our speaker-adaptive post-processing. In accordance with the shared-task rules, we used only the officially released training data and no external resources.

4. System Description

4.1. Overview

Our system followed a multi-stage pipeline for Arabic speech dictation with automatic diacritization. First, each input utterance was transcribed offline using a Whisper-based ASR model (Shatnawi, 2024), and the resulting transcripts were cached. Second, we applied *ASR Consistency Filtering* to determine whether each ASR transcript should be used directly or replaced with a fallback representation during training. Third, we fine-tuned a pre-trained character-level diacritization model under

three runs: two with different filtering thresholds ($\tau=0.5$ and $\tau=0.9$) and one additional run at $\tau=0.5$ with a different random seed to increase ensemble diversity. Finally, at inference time, we combined the three checkpoints with confidence-based ensemble decoding and applied speaker-adaptive post-processing with constrained decoding to produce the final diacritized output.

4.2. Base Model and ASR

We built on the Diac Transformer, a character-level Transformer encoder with 1,631,507 trainable parameters, $d_{\text{model}} = 256$, and two Transformer blocks. The model used two input branches: a text branch encoding undiacritized Arabic input and an ASR branch encoding a diacritized Whisper transcript. The two branches were fused through cross-attention inside the encoder. A classification head predicted one diacritic label per character, where label 0 denoted the absence of a diacritic. We initialized the model from the pretrained checkpoint `rufaelfekadu/diac-transformer-text-asr-tashkeela-clartts`¹ (Fekadu, 2024).

ASR transcripts were generated using `sashat/whisper-medium-ClassicalAr`² (Shatnawi, 2024), used as a pretrained component without further fine-tuning. Train, development, and test ASR outputs were cached and reused during training and inference. Spoken punctuation was removed from ASR transcripts before passing them to the ASR branch.

4.3. Preprocessing and ASR Consistency Filtering

During TSV construction, rows were skipped if the diacritized gold text contained no Arabic letters, contained no harakat, or became empty after sanitization, where sanitization replaces tabs and line breaks with spaces, collapses repeated spaces, and strips surrounding whitespace.

To reduce the effect of noisy ASR inputs, we introduced *ASR Consistency Filtering*. For each example i , we computed a similarity ratio between the space-removed undiacritized gold text and the space-removed diacritics-stripped ASR output. Let a_i denote the space-removed undiacritized gold text and b_i the space-removed diacritics-stripped ASR output. The consistency score was computed as

$$\text{ratio}_i = \text{SequenceMatcher}(a_i, b_i). \quad (1)$$

¹<https://huggingface.co/rufaelfekadu/diac-transformer-text-asr-tashkeela-clartts>

²<https://huggingface.co/sashat/whisper-medium-ClassicalAr>

Given a threshold τ , the ASR-side input used during training, denoted by $\hat{x}_i$, was determined as follows:

$$\hat{x}_i = \begin{cases} r_i & \text{if } \text{ratio}_i \geq \tau, \\ \bar{x}_i & \text{if } \text{ratio}_i < \tau \text{ or } r_i \text{ is empty,} \end{cases} \quad (2)$$

where r_i is the punctuation-cleaned ASR transcript and $\bar{x}_i$ is the undiacritized gold text. Thus, unreliable ASR was replaced with the undiacritized gold text as a fallback input rather than removed. This uses the undiacritized gold text rather than diacritized gold, so it does not inject target diacritics into the ASR branch during training.

We trained two threshold settings, $\tau = 0.5$ (`tau05`) and $\tau = 0.9$ (`tau09`). In addition, we trained a third run at $\tau = 0.5$ with a different random seed (`seed123`) to increase ensemble diversity. All three checkpoints were combined in the final ensemble.

4.4. Fine-Tuning

We fine-tuned the model using Adam with a learning rate of 5×10^{-5} and weight decay of 10^{-4} . Training used cross-entropy loss with label smoothing of 0.1, batch size 8, gradient accumulation over 16 steps, gradient clipping with a maximum norm of 1.0, and a cosine annealing learning-rate schedule with $\eta_{\min} = 10^{-6}$. We trained for up to 500 epochs and applied early stopping with a patience of 70 based on validation loss. The hyperparameters were selected based on development-set validation loss and WER.

We trained three runs: `tau05` ($\tau = 0.5$, default seed), `tau09` ($\tau = 0.9$, default seed), and `seed123` ($\tau = 0.5$, seed 123). The additional `seed123` run was included to increase ensemble diversity. We did not evaluate multiple alternative random seeds. Notably, `tau05` and `seed123` used the same filtered TSVs and differed only in random initialization. For all runs, we saved only the checkpoint with the best validation loss and used it for inference.

4.5. Ensemble and Post-Processing

At inference time, we combined the three checkpoints (`tau05`, `tau09`, and `seed123`) using character-level confidence voting. For each character position i , we computed softmax probabilities for each model and selected the prediction from the model with the highest maximum probability at that position:

$$\begin{aligned} k^* &= \arg \max_k \max(\text{probs}_k[i]), \\ \hat{y}_i &= \arg \max(\text{probs}_{k^*}[i]). \end{aligned} \quad (3)$$

This ensemble operated position-wise and did not average probabilities across models.

θ	DER	WER	#SUKUN
0.55	10.04	32.63	15
0.60	9.77	31.26	13
0.65	9.76	31.18	9
0.70	9.75	31.19	12
0.75	9.99	31.72	7
0.80	10.16	32.02	4

Table 1: Grid search for the speaker-adaptive threshold θ on the development set (WCE, Incl. 0). Lower is better for DER and WER.

We then applied speaker-adaptive post-processing. Using development-set predictions, we counted word-final sukun versus vowel realizations for each speaker and classified speakers as `SUKUN`, `VOWEL`, or `MIXED` using a threshold θ . Intuitively, `SUKUN` speakers tend to suppress word-final case endings, `VOWEL` speakers tend to preserve them, and `MIXED` speakers show no strong preference. During post-processing, word-final tanwin and short case vowels were replaced with sukun only for `SUKUN` speakers; predictions for `VOWEL` and `MIXED` speakers were left unchanged. Speakers unseen in the development set were treated as `MIXED`.

We selected θ by grid search over $\{0.55, 0.60, 0.65, 0.70, 0.75, 0.80\}$ on the development set using WER as the selection criterion, since WER requires the full word to be diacritized correctly. As shown in Table 1, the best development-set WER was obtained at $\theta = 0.65$, which was therefore used in the final submitted system.

Finally, we applied constrained decoding after speaker-adaptive post-processing: if the diacritics-stripped predicted word did not match the diacritics-stripped input word, the predicted word was replaced with the original undiacritized input token. This served as a safeguard against word-level mismatches.

5. Results and Analysis

We report Diacritic Error Rate (DER), Word Error Rate (WER), and Sentence Error Rate (SER). Following the shared-task evaluation, we treat WER as the primary metric because it requires the full word to be diacritized correctly.

5.1. Development Set Ablation

Table 2 shows the contribution of each system component on the development set, evaluated both with and without case endings. Fine-tuning on consistency-filtered data provided the largest improvement, reducing WER from 43.07 to 32.75 with `tau05`. The stricter filtering run `tau09` ($\tau=0.9$) performed slightly worse than `tau05`, suggesting that

Setting	System	DER	WER	SER
WCE	Baseline	15.94	43.07	85.38
	$\tau=0.5$	10.28	32.75	85.00
	$\tau=0.9$	10.79	33.38	87.31
	seed123	10.34	32.45	87.21
	Ensemble	10.24	32.15	84.23
	+Post-proc	9.72	31.23	83.46
WOCE	Baseline	10.21	24.86	78.08
	$\tau=0.5$	6.37	18.01	70.00
	$\tau=0.9$	6.82	18.51	71.54
	seed123	6.32	17.51	70.60
	Ensemble	6.29	17.58	69.62
	+Post-proc	6.29	17.58	69.62

Table 2: Development-set ablation results under the with-case-ending (WCE) and without-case-ending (WOCE) settings. Ensemble combines $\tau=0.5$, $\tau=0.9$, and seed123. Post-processing uses $\theta = 0.65$. Lower is better.

aggressive fallback reduced the usefulness of ASR-side information. Ensembling the three checkpoints yielded a further but modest gain, improving WER from 32.75 to 32.15. Finally, speaker-adaptive post-processing with $\theta=0.65$ and constrained decoding reduced WER by an additional 0.92 points, producing the best development result of **9.72 DER, 31.23 WER, and 83.46 SER**. Under the WOCE metric, post-processing yielded no additional gain, as the speaker-adaptive corrections were confined to case endings, which are excluded from WOCE evaluation.

5.2. Official Test Results

Table 3 compares our official submission against the organizers’ fine-tuned Text+ASR baseline. Our system outperformed the baseline in three of the four official evaluation settings. Under Incl. 0, WCE, our system improved WER from 31.84 to 30.37 while also reducing DER and SER. Under Incl. 0, WOCE, we improved all three metrics, including DER from 7.89 to 6.76. Under Excl. 0, WCE, the gains were smaller but still consistent across all three metrics. The only setting where the baseline remained stronger was Excl. 0, WOCE, which is consistent with our design because the post-processing mainly targets word-final case-ending behavior.

5.3. Analysis

Effect of the consistency threshold. In our experiments, the milder setting $\tau=0.5$ performed better than the stricter setting $\tau=0.9$: $\tau=0.5$ achieved 10.28 DER and 32.75 WER on the development set, compared with 10.79 DER and 33.38 WER for $\tau=0.9$. This suggests that overly aggressive fallback weakens the benefit of the ASR branch by

Setting	System	DER	WER	SER
Incl. 0, WCE	Baseline	9.91	31.84	82.93
	Ours	8.23	30.37	80.79
Incl. 0, WOCE	Baseline	7.89	20.99	67.07
	Ours	6.76	20.84	64.02
Excl. 0, WCE	Baseline	8.52	24.73	78.66
	Ours	8.46	24.33	74.39
Excl. 0, WOCE	Baseline	4.82	10.89	50.61
	Ours	5.87	13.43	55.18

Table 3: Official test-set results for our submission compared with the organizer’s fine-tuned Text+ASR baseline. WCE = with case endings, WOCE = without case endings, and 0 denotes no-diacritic characters. Lower is better.

discarding useful speech-side information together with noisy transcriptions.

Effect of ensembling. The three-checkpoint confidence-voting ensemble improved development performance from 32.75 to 32.15 WER and from 10.28 to 10.24 DER. This suggests limited but useful complementarity among the checkpoints.

Effect of speaker-adaptive post-processing. Speaker-adaptive post-processing produced the final development gain, reducing WER from 32.15 to 31.23 and DER from 10.24 to 9.72. The largest benefit was observed in settings where case endings remained part of the evaluation, consistent with the design of the post-processing step. In practice, these gains were most visible in word-final positions, where over-predicted case vowels were corrected to sukun for speakers with sukun-dominant realizations.

Remaining error sources. The remaining errors appear to be concentrated in word-final diacritics. Our system outperformed the organizers’ fine-tuned Text+ASR baseline in three of four test settings, but underperformed it in the Excl. 0, WOCE condition. This suggests that our strongest gains came from correcting case-ending behavior, while other diacritic errors remain more challenging.

6. Conclusion

We presented our submission to Task 2 of the KSAA-2026 Shared Task on Arabic Speech Dictation with Automatic Diacritization. Building on the Diac Transformer, our system combined ASR Consistency Filtering, confidence-based ensembling, speaker-adaptive post-processing, and constrained decoding. On the official test set, it achieved 8.23 DER and 30.37 WER under Incl. 0, WCE. Limitations include dependence on development-set speaker statistics, sensitivity to the filtering threshold, reduced transfer across ASR models or dialects, and the absence of explicit syntactic modeling.

7. Bibliographical References

Asma Al Wazrah, Waad Alshammari, Rawan Almatham, Raghad Al-Rasheed, Afrah Altamimi, Rufael Marew, Sawsan Alqahtani, Hanan Aldarmaki, Abdullah Alharbi, Abdulrahman Alshehri, Mohamed Assar, Amal Almazrui, and Abdulrahman AlOsaimy. 2026. Ksaa-2026 shared task on arabic speech dictation with automatic diacritization. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*.

Hanan Aldarmaki and Ahmad Ghannam. 2023. [Diacritic recognition performance in arabic asr](#). pages 361–365.

Badr AlKhamissi, Muhammad ElNokrashy, and Mohamed Gabr. 2020. [Deep diacritization: Efficient hierarchical recurrence for improved Arabic diacritization](#). In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 38–48, Barcelona, Spain (Online). Association for Computational Linguistics.

Salman Elgamal, Ossama Obeid, Mhd Kabbani, Go Inoue, and Nizar Habash. 2024. [Arabic diacritics in the wild: Exploiting opportunities for improved diacritization](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14815–14829, Bangkok, Thailand. Association for Computational Linguistics.

Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019. [Neural Arabic text diacritization: State of the art results and a novel approach for machine translation](#). In *Proceedings of the 6th Workshop on Asian Translation*, pages 215–225, Hong Kong, China. Association for Computational Linguistics.

Hamdy Mubarak, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. 2019. [Highly effective Arabic diacritization using sequence to sequence modeling](#). 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 2390–2395, Minneapolis, Minnesota. Association for Computational Linguistics.

Han Qin, Guimin Chen, Yuanhe Tian, and Yan Song. 2021. [Improving Arabic diacritization with regularized decoding and adversarial training](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and*

the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 534–542, Online. Association for Computational Linguistics.

Sara Shatnawi, Sawsan Alqahtani, and Hanan Aldarmaki. 2024. [Automatic restoration of diacritics for speech data sets](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4166–4176, Mexico City, Mexico. Association for Computational Linguistics.

Brian Thompson and Ali Alshehri. 2022. [Improving Arabic diacritization by learning to diacritize and translate](#). In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 11–21, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

8. Language Resource References

Fekadu, Rufael. 2024. [Diac Transformer: Text and ASR-based Arabic Diacritization Model](#). Hugging Face model repository.

Shatnawi, Sara. 2024. [Whisper Medium Classical Arabic](#). Hugging Face model repository.

Abjad AI at KSAA-2026 Shared Task 2: Grouped Speech Conditioning for Arabic Diacritization

Naif Alharthi, Ahmad Ghannam, Faris Alasmay, Kholood Al Tabash, Shouq Sadah, Lahouari Ghouti

Abjad AI, Saudi Arabia

{nalharthi, aghannam, falasmay, kaltabash, ssadah, lghouti}@abjad.com.sa

Abstract

We describe Abjad AI’s submission to KSAA-2026 Shared Task 2 on automatic diacritization of Arabic speech dictation. The task requires generating fully diacritized text given speech audio and an undiacritized transcript. Because text-only diacritization cannot resolve ambiguities that are recoverable from the acoustic signal, we propose conditioning a character-level encoder-only Transformer (CATT) on speech representations. We introduce grouped speech conditioning, which downsamples speech encoder features into a small set of pooled tokens concatenated to the text input, enabling efficient fusion without architectural changes to CATT. We train with a two-phase schedule that first freezes the text encoder, then fine-tunes the full model. Our best system, using Whisper-small features with grouping size 5, achieves a Diacritization Error Rate (DER) of 5.68 and a Word Error Rate (WER) of 11.85 under the official primary test setting. Notably, we find that Whisper-small consistently outperforms Whisper-large-v3, suggesting that compact speech representations better suit this fusion setting. This is an extended and revised version of our previous work (Ghannam et al., 2025).

Keywords: Arabic diacritization, multimodal learning, speech conditioning, Whisper, KSAA shared task

1. Introduction

Arabic diacritics (tashkīl) are typically omitted in everyday writing, which increases lexical ambiguity and negatively impacts applications such as text-to-speech, reading assistance, and educational tools. Automatic diacritization aims to restore missing diacritics, but purely text-based systems struggle in cases where multiple vowelizations are plausible without additional context. In dictation settings, audio provides strong phonetic cues that can resolve ambiguity and guide diacritic placement.

KSAA-2026 Shared Task 2 targets *speech-aware* diacritization: given an audio clip and its undiacritized transcript, systems must output fully diacritized Arabic. Our system continues the direction of our earlier CATT-Whisper work (Ghannam et al., 2025), with a focus on a lightweight fusion mechanism and a systematic sweep over speech backbones and speech-to-text conditioning strength.

Contributions. In this paper, we describe our system and report three main findings: (1) A simple concatenation-based fusion strategy, where speech features are average-pooled over consecutive groups of G frames and prepended to the text sequence, is sufficient to improve diacritization over a text-only baseline without requiring cross-attention or other expensive mechanisms. (2) Whisper-small (Radford et al., 2022) consistently outperforms both smaller and larger Whisper variants as well as Squeezeformer (Kim et al., 2022), suggesting diminishing returns from increased encoder capacity in this setting. (3) A two-stage train-

ing schedule, which freezes the text encoder for the first 10 epochs before unfreezing, stabilizes training, though we note that this choice was not ablated against alternative schedules.

2. Task and Data

The task input consists of speech audio and undiacritized transcript pairs; the output is the diacritized transcript. We follow the official shared-task split and evaluation setup. We apply minimal text normalization: removing non-Arabic characters and collapsing whitespace. We do not use data augmentation.

2.1. Related Work

Recent work on Arabic diacritic restoration (DR) has increasingly explored multimodal approaches that incorporate speech information to address ambiguities that cannot be resolved from text alone (Elgamal et al., 2024). Because Arabic orthography typically omits short vowels, many surface forms admit multiple valid vowelizations, which makes acoustic evidence particularly valuable in speech-driven applications such as dictation and automatic speech recognition (ASR).

Speech-aware DR uses acoustic features extracted from audio to improve diacritization accuracy. Previous work has shown that speech information is especially helpful in ASR and dictation tasks, where pronunciation cues can help select the correct vowelization, particularly for dialectal and spontaneous speech (Shatnawi et al., 2024a,b).

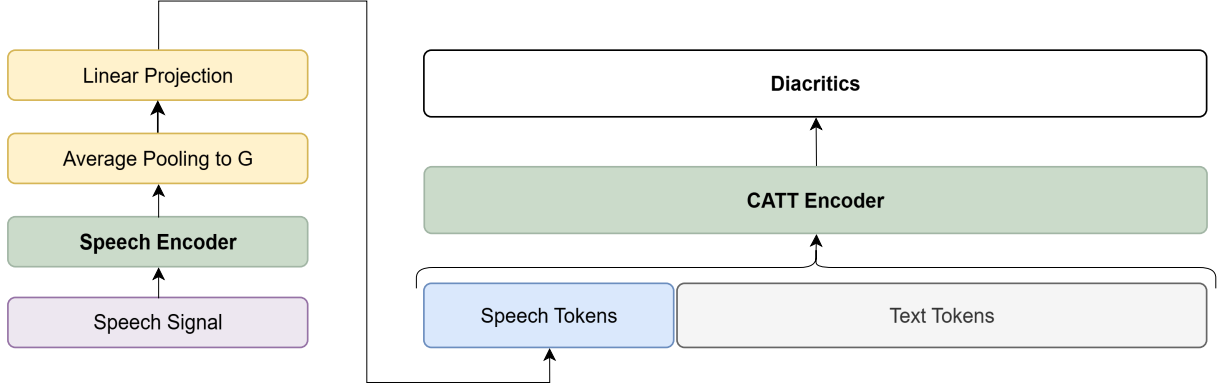


Figure 1: Overview of our grouped speech conditioning architecture for Arabic diacritization. The speech signal is encoded by a speech backbone (Whisper/Squeezeformer) to produce T frame-level vectors, which are average-pooled over consecutive groups of G frames. The pooled representations are then linearly projected to the CATT embedding dimension, concatenated with the text tokens, and fed into the CATT encoder to predict the output diacritics.

Multimodal fusion architectures have been proposed to combine speech and text representations for DR. Our prior work, CATT-Whisper, demonstrated that integrating a Whisper speech encoder with a character-level diacritization model improves performance by injecting phonetic information into the prediction process (Ghannam et al., 2025).

3. Methodology

In this work, we extend our previous CATT-Whisper model by improving the multimodal fusion between speech and text through a lightweight grouped speech conditioning mechanism. We also compare different speech encoders, including Whisper and Squeezeformer, to study the trade-offs between efficiency and diacritization performance in the KSAAs dictation setting.

3.1. Text Backbone: CATT

We use CATT, a character-level encoder-only Transformer (char-BERT style) (Alasmay et al., 2024). Given a character sequence, CATT produces contextualized representations and a token-level classifier predicts diacritic tags per character. This architecture supports fine-grained diacritics and is compatible with long sequences.

3.2. Speech Backbone

We evaluate two speech encoders:

- **Whisper**: encoder representations from Whisper variants (base, small, large-v3) (Radford et al., 2022).
- **Squeezeformer**: an efficient Transformer-based ASR backbone (Kim et al., 2022).

3.3. Grouped Speech Conditioning

We use grouping to control how much speech information is injected while keeping compute costs low. We tune the grouping factor $G \in \{3, 5, 10, 15\}$, where G denotes the number of consecutive frame-level vectors averaged into one pooled speech token. Let $S \in \mathbb{R}^{T \times d_s}$ be frame-level speech representations and $X \in \mathbb{R}^{L \times d_t}$ be text embeddings. We downsample S by average-pooling every G frames, producing $\tilde{S} \in \mathbb{R}^{K \times d_s}$ where $K = T/G$. We then linearly project to the text dimension and concatenate:

$$Z = [\text{Proj}(\tilde{S}); X] \in \mathbb{R}^{(K+L) \times d_t}.$$

The concatenated sequence Z is fed to the CATT encoder. A token-level classifier predicts diacritics for the text positions. Too small G retains a long speech-token sequence ($K = T/G$), which can inject excessive and potentially redundant acoustic context and increase compute due to the longer concatenated sequence ($K + L$). Conversely, too large G over-compresses speech by averaging many frames into each token, reducing K and compute but potentially smoothing away short-vowel cues and other fine-grained phonetic information needed for accurate diacritization. We therefore tune G on the development set to balance acoustic detail against sequence length and efficiency.

4. Training Setup

We train with cross-entropy over diacritic labels using AdamW (Loshchilov and Hutter, 2019), batch size 32, and maximum length 1024. For all experiments, we initialize the CATT text backbone from a pretrained encoder-only char-BERT checkpoint,

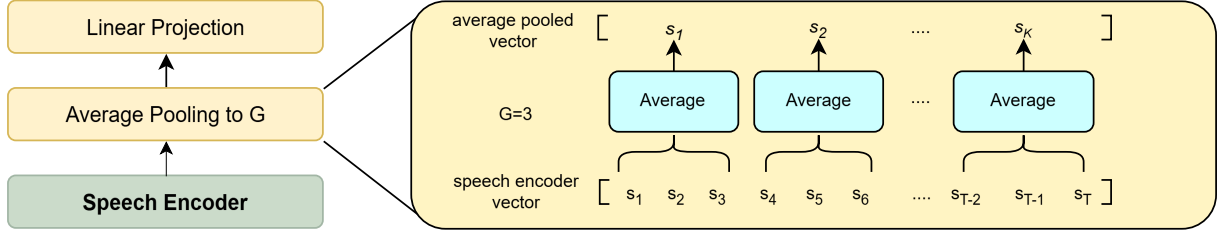


Figure 2: Grouped speech conditioning ($G=3$). Frame-level speech embeddings are grouped into chunks of three and average-pooled into pooled speech tokens ($s_1, s_2, \dots, s_K$), then linearly projected to the CATT embedding space to condition the diacritization model.

while the speech encoder is initialized from the selected backbone (Whisper or Squeezeformer). We select hyperparameters on the development set using the official primary metric setting (WOCE, excluding no-diacritic), and we report the full metric grid in Table 1. We use a two-stage schedule:

1. **Freeze CATT** for the first 10 epochs, allowing the speech projection and fusion to adapt.
2. **Unfreeze CATT** and continue for 10 additional epochs.

Freezing the pretrained CATT backbone initially stabilizes training by preventing early updates from disrupting text representations. This warm-up lets the model learn a mapping from speech tokens to the CATT space for cross-modal conditioning. In the second stage, unfreezing CATT allows end-to-end adaptation, letting the text encoder specialize for dictation and better integrate acoustic cues for diacritics.

5. Evaluation Protocol

We use the same evaluation protocol used in the baseline, reporting WER and DER under: (i) **with case ending** vs. (ii) **without case ending**; and (iii) **including no-diacritic** vs. (iv) **excluding no-diacritic**¹. Unless otherwise stated, we highlight the *without case ending* setting.

6. Results

6.1. Development Set: Backbone and Group Size Sweep

Table 1 summarizes the dev set sweep over speech backbones and grouping size G using the selected settings, excluding no-diacritic tokens and without case endings.

Backbone	Metric	Group size G			
		3	5	10	15
Sqf	DER	6.91	6.82	6.84	7.01
	WER	14.80	14.60	14.60	15.07
W-base	DER	6.92	6.64	6.56	6.42
	WER	14.80	14.32	14.08	13.80
W-small	DER	6.39	6.18	6.40	6.32
	WER	13.53	13.29	13.53	13.61
W-large	DER	6.75	6.50	6.56	6.61
	WER	14.68	14.16	14.12	14.28

Table 1: Development sweep (Sqf for Squeezeformer, W for Whisper).

6.2. Full Dev Metrics for the Best Model

	Incl		Excl	
	DER	WER	DER	WER
Whisper-small, $G=5$				
WCE	8.15	28.32	9.08	26.14
WOCE	4.62	15.76	6.18	13.29

Table 2: Full evaluation metrics for Whisper-small with $G = 5$ on the development set.

6.3. Shared Task Test Results

Participant	DER	WER
meshal	4.80	10.48
nadaadelmoussa	5.15	11.06
naif_alharthi	5.68	11.85
nahian_abu	5.87	13.43
Hassan	6.86	14.32
omarnj10	25.27	31.07
astral_fate	25.03	49.07

Table 3: Official test leaderboard (Al Wazrah et al., 2026). Our values are bolded.

¹<https://github.com/rufaelfekadu/Diac>

7. Discussion

We observe three consistent trends. First, **Whisper backbones outperform Squeezeformer** across all grouping factors, suggesting that Whisper’s large-scale pretraining provides more transferable acoustic representations for dictation-style diacritization. Second, the **grouping factor G (frames averaged per pooled token)** offers an accuracy–efficiency knob: smaller G yields more pooled speech tokens ($K = T/G$) and longer sequences ($K + L$), which can add redundant context and increase compute, while larger G compresses speech more aggressively and may smooth away short-vowel cues. Empirically, Whisper-small peaks at $G=5$ on the dev set, indicating a good balance between acoustic detail and compression. Third, results under **excluding no-diacritic** provide a sharper signal for model selection than the including setting, since they focus on positions where diacritics matter. These findings align with prior speech-based diacritization work highlighting the value of acoustic cues and robustness techniques such as augmentation (Shatnawi et al., 2024a,b).

8. Conclusion

We presented Abjad AI’s system for KSAA-2026 Shared Task 2: a character-level diacritization model conditioned on speech via grouped speech tokens. Whisper-small with $G=5$ performed best in our experiments.

9. Bibliographical References

- Asma Al Wazrah, Waad Alshammari, Rawan Almatham, Raghad Al-Rasheed, Afrah Altamimi, Rufael Marew, Sawsan Alqahtani, Hanan Aldarmaki, Abdullah Alharbi, Abdulrahman Alshehri, Mohamed Assar, Amal Almazrui, and Abdulrahman AlOsaimey. 2026. Ksaa-2026 shared task on arabic speech dictation with automatic diacritization. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7)*.
- Faris Alasmay, Orjuwan Zaafarani, and Ahmad Ghannam. 2024. [CATT: Character-based Arabic tashkeel transformer](#). In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 250–257, Bangkok, Thailand. Association for Computational Linguistics.
- Salman Elgamal, Ossama Obeid, Mhd Kabbani, Go Inoue, and Nizar Habash. 2024. [Arabic diacritics in the wild: Exploiting opportunities for improved diacritization](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14815–14829, Bangkok, Thailand. Association for Computational Linguistics.
- Ahmad Ghannam, Naif Alharthi, Faris Alasmay, Kholood Al Tabash, Shouq Sadah, and Lahouari Ghouti. 2025. [Abjad AI at NADI 2025: CATT-whisper: Multimodal diacritic restoration using text and speech representations](#). In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 757–761, Suzhou, China. Association for Computational Linguistics.
- Sehoon Kim, Amir Gholami, Albert Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W. Mahoney, and Kurt Keutzer. 2022. [Squeezeformer: An efficient transformer for automatic speech recognition](#). arXiv preprint arXiv:2206.00888.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). arXiv preprint arXiv:1711.05101.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. [Robust speech recognition via large-scale weak supervision](#). arXiv preprint arXiv:2212.04356.
- Sara Shatnawi, Sawsan Alqahtani, and Hanan Aldarmaki. 2024a. [Automatic restoration of diacritics for speech data sets](#). arXiv preprint arXiv:2311.10771.
- Sara Shatnawi, Sawsan Alqahtani, Shady Shehata, and Hanan Aldarmaki. 2024b. [Data augmentation for speech-based diacritic restoration](#). In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 160–169, Bangkok, Thailand. Association for Computational Linguistics.

AraSentEval 2026: A Shared Task on Sentiment Analysis and Swapping in Arabic

Saad Ezzini¹, Shadi Abudalfa¹, Maram Alharbi², Salmane Chafik³,
Hamzah Luqman¹, Mo El-Haj^{2,4}, Paul Rayson^{2,4}, Samaher Alghamdi², Reem Alotaibi⁵

¹King Fahd University of Petroleum & Minerals, KSA

²Lancaster University, UK

³Mohammed VI Polytechnic University, Morocco

⁴VinUniversity, Vietnam

⁵King Abdulaziz University, KSA

Abstract

Sentiment analysis is a fundamental problem in Natural Language Processing (NLP). Standard sentiment classification for the Arabic language remains challenging due to the high volume of dialectal Arabic. To advance research in this area, this paper proposes the Shared Task on Sentiment Analysis and Swapping in Arabic (AraSentEval), organized as part of the OSACT7 Workshop at LREC 2026. This shared task consists of two subtasks: Subtask 1 focuses on multi-class and multi-dialect sentiment analysis, requiring models to identify sentiment polarity across various Arabic dialects. Subtask 2 introduces a generative task for Arabic sentiment swap, challenging models to invert sentiment polarity while preserving core semantics. In this overview paper, we present the motivation, dataset creation, and summarize the main findings from participating models.

Keywords: Arabic NLP, Sentiment Analysis, Text Style Transfer, Shared Task

1. Introduction

Sentiment analysis remains a key task in Natural Language Processing (NLP), with critical real-world applications ranging from social media monitoring to customer feedback analysis. While significant progress has been made for many languages, and multiple types have been defined (document, sentence, and feature/aspect level), the Arabic language presents unique challenges due to its rich morphology and extensive dialectal variation. Most user-generated content, the primary source for sentiment data, is written in dialectal Arabic, which is often under-resourced and differs significantly from Modern Standard Arabic (MSA).

To address these challenges and foster innovation in Arabic NLP, we propose a novel Shared Task on Sentiment Analysis and Swapping in Arabic (AraSentEval), hosted at the OSACT7 Workshop at LREC 2026¹. This task is designed to move beyond standard sentiment classification by evaluating both the understanding and generation of sentiment in diverse Arabic contexts. AraSentEval comprises two distinct but complementary subtasks:

- **Subtask 1: Arabic Dialect Sentiment Analysis:** A multi-class classification task focused on identifying the sentiment (positive, negative, or neutral) of texts written in various Arabic dialects. This subtask emphasizes the need for robust models that can handle the lexical and syntactic diversity of the Arab world.

- **Subtask 2: Arabic Sentiment Swap:** A generative task where participants' systems must rewrite a given sentence to invert its sentiment polarity (e.g., positive to negative) while preserving its core meaning. This challenges models to go beyond surface-level cues and demonstrate a deeper grasp of semantics and syntax.

2. Motivation

The motivation for AraSentEval is grounded in both linguistic and methodological gaps in current Arabic NLP research.

First, despite sustained interest in sentiment analysis for Arabic, there remains a clear shortage of robust benchmarks that adequately capture dialectal diversity. Earlier shared tasks and evaluation campaigns (El-Beltagy et al., 2017; Rosenthal et al., 2017) have advanced sentiment classification for Arabic, yet most resources are either limited to Modern Standard Arabic (MSA) or focus on a narrow set of dialects and domains (El-Haj, 2026, 2020). In practice, however, the majority of user-generated content, including reviews, social media posts, and online discussions, is written in dialectal Arabic. These dialects exhibit substantial lexical, morphological, and syntactic variation, which often leads to performance degradation when models trained on MSA are applied to real-world data. Prior work has highlighted the fragmentation of Arabic sentiment resources and the difficulty of transferring models across dialects (Nwesri et al., 2025; Alwakid et al.,

¹<https://ezzini.github.io/AraSentEval/>

2022), reinforcing the need for more representative evaluation benchmarks. By introducing a balanced, multi-dialect benchmark informed by recent benchmarking efforts (Alharbi et al., 2025c), AraSentEval seeks to promote the development of sentiment models that generalise across dialects and better reflect authentic language use in the Arab world.

Second, the task aims to extend evaluation beyond classification into controlled text generation. While sentiment classification has reached a relatively mature stage in Arabic NLP, generative sentiment manipulation remains underexplored. Sentiment swapping, framed as a form of style transfer, requires systems to invert polarity while preserving core propositional content. This problem has been investigated in English (Shen et al., 2017; Li et al., 2018), where style transfer research has demonstrated the complexity of disentangling content from sentiment and other stylistic attributes. Comparable resources and systematic evaluation frameworks are scarce for Arabic, limiting progress in controllable generation (Abudalfa et al., 2024; Abdu et al., 2025). The limited availability of high-quality parallel datasets has constrained progress in this area. By introducing a dedicated sentiment swap subtask, AraSentEval encourages research on semantic preservation, polarity control, and evaluation methodologies for Arabic natural language generation. Such capabilities have practical implications for data augmentation, controllable dialogue systems, and adaptive content generation, particularly in low-resource and dialectally diverse settings.

By combining a dialect-aware classification task with a challenging generative task, AraSentEval offers a unified benchmark for both sentiment understanding and sentiment manipulation in Arabic. This dual focus is intended to stimulate research that moves beyond surface-level polarity detection towards more semantically grounded and controllable language technologies for Arabic.

3. Data Collection and Creation

The datasets for both subtasks were collected and annotated prior to the task launch. Figure 1 illustrates the dataset creation process for both subtasks.

3.1. Subtask 1: Dialect Sentiment Analysis

The dataset for this task, Multi-Dialect-Sent (MDS-3), currently consists of 3,000 sentences annotated for three-class sentiment (positive, negative, neutral). The dataset is balanced across four major dialects: Moroccan, Egyptian, Jordanian, and Saudi. The data was sourced from hotel reviews from the

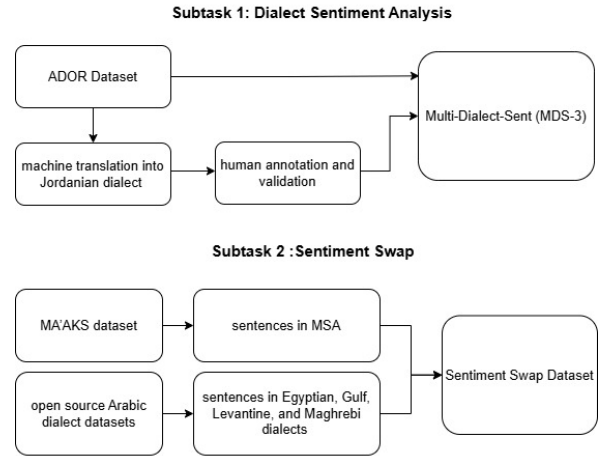


Figure 1: Data creation process for both subtasks.

ADOR dataset (Alharbi et al., 2025b), then translated and annotated by native speakers of each dialect. The sentences are parallel across dialects, with balanced sentiment representation across all included dialects. An initial version of the dataset was verified in a recently organised shared task Ahasis (Alharbi et al., 2025a) at RANLP-2025.

The dataset was split into 80% for training and 20% for testing. The split was performed based on sentence IDs to ensure that parallel sentences remain in the same set, to prevent potential data leakage between the training and test sets.

3.2. Subtask 2: Sentiment Swap

The data employed in this subtask is primarily derived from the MA'AKS dataset (Mughaus et al., 2026), a manually constructed parallel resource designed for Arabic sentiment swap. This dataset contains 5,000 paired sentences in MSA. Within each pair, both sentences address the same underlying subject matter but express contrasting sentiment orientations—one conveying a positive stance and the other a negative one. The dataset was assembled from publicly accessible resources and underwent careful manual annotation and cross-checking to guarantee linguistic fluency, semantic consistency, and precise sentiment transformation.

For experimental purposes, MA'AKS dataset is partitioned into three subsets: 4,000 pairs for training, 500 for validation, and 500 for testing. It has been evaluated using multiple advanced Large Language Models (LLMs), such as AceGPT, JAIS, and Llama-3, under different configurations including zero-shot prompting, few-shot prompting, and supervised fine-tuning. These evaluations demonstrate both the robustness of the dataset and the practicality of the sentiment-swapping task.

Beyond MA'AKS dataset, the subtask also incorporates additional data representing several

Arabic dialects, namely Egyptian, Gulf, Levantine, and Moroccan varieties. These dialectal examples were sourced from other open datasets and subsequently annotated and verified with the assistance of open-source LLMs. Following this integration and validation process, the final dataset for the subtask comprises 6,263 MSA and dialectal pairs in the training split, 1,315 pairs in the development split, and 646 dialect-only pairs in the test split. The test set was intentionally restricted to dialectal Arabic to rigorously evaluate the systems' generalization abilities to dialectal variations, excluding the relatively easier MSA content.

4. Task Description and Evaluation

In this section, we describe in detail the subtasks that define our shared task. Participants were allowed to participate in either one or both subtasks. The official evaluation was conducted on the **CodaBench** platform, an open-source platform designed to organize Artificial Intelligence benchmarks by defining tasks, including datasets, submission pipelines, and evaluation metrics. It provides an automated and transparent framework for evaluating system submissions.

4.1. Subtask 1: Arabic Dialect Sentiment Analysis

In this subtask, participants are required to build a system that classifies the sentiment expressed in sentences written in four different and linguistically challenging Arabic dialects: Moroccan (Dar-ija), Egyptian, Jordanian, and Saudi. The goal of this task is to evaluate the ability of models to handle dialectal variation in Arabic while accurately identifying the sentiment conveyed in each sentence.

Formally, the objective is to design a sentence-level classifier C such that, given an input sentence x , the system predicts a sentiment label S , where:

$$S = C(x), \quad S \in \{\text{Positive, Neutral, Negative}\}.$$

- **Input:** A sentence written in Arabic using one of the target dialects (Moroccan, Egyptian, Jordanian, or Saudi).
- **Output:** A sentiment label selected from the set {Positive, Neutral, Negative}.
- **Evaluation Metric:** Systems are evaluated using the **Macro F1-score**. This metric is chosen because it gives equal importance to all classes and is therefore more suitable in scenarios where class distributions may be imbalanced.

4.2. Subtask 2: Arabic Sentiment Swap

In the second subtask, participants are expected to design models able to generate a new sentence that expresses the opposite sentiment of a given sentence while preserving its original semantic content. The input sentences are written in either Modern Standard Arabic (MSA) or in one of several Arabic dialects, including Egyptian, Gulf, Levantine, and Moroccan. This task evaluates a model's ability to perform controlled sentiment transformation while maintaining semantic fidelity.

Formally, the objective is to train a model M capable of performing sentiment-controlled text generation. Given an input sentence x expressing a sentiment S , the model should generate a new sentence x' that conveys the opposite sentiment S' , while preserving the core meaning and topic of the original sentence, such that:

$$x' = M(x), \quad S' = \text{Opposite}(S)$$

where $S, S' \in \{\text{Positive, Negative}\}$ and $S' \neq S$.

- **Input:** An Arabic sentence along with its original sentiment polarity (e.g., "هذا المطعم رائع للغاية", Positive).
- **Output:** A rewritten Arabic sentence that preserves the main meaning of the input while expressing the opposite sentiment polarity (e.g., "هذا المطعم سيء للغاية").
- **Evaluation Metrics:** The generated outputs are evaluated using both sentiment accuracy and text similarity metrics to ensure correct sentiment transfer while maintaining semantic consistency.
 - **Sentiment Style Accuracy:** A pre-trained sentiment classifier is used to measure the percentage of generated sentences that correctly express the target polarity.
 - **BLEU and chrF:** These metrics are used to evaluate the degree of lexical and character-level similarity between the generated sentence and the reference sentence, providing an indication of semantic preservation.

5. Participant Systems

A total of 15 teams participated in the shared task, of which 9 submitted system description papers for AraSentEval. Below is a brief summary of the methodologies proposed by the participating teams, mapping their team names to their corresponding leaderboard usernames, their chosen subtask, and their reported scores and rankings:

- **musj1984** (Username: *musj1984*): Participated in **Subtask 1**. They adapted a Mixture of Experts (MoE) framework to handle Arabic three-class sentiment classification by combining two QLoRA fine-tuned models (AraBERT v2 and BERT-Large Arabic) using a learned gating network. Training data was enriched via LLM-based augmentation. As reported in their paper, they achieved a Macro F1-score of 0.9423, ranking **1st** overall in the subtask.
- **TTLab** (Username: *alighabusaleh*): Participated in **Subtask 1**. They proposed SARF, a multi-view architectural framework integrating surface-level context with stemmed and rooted morphological perspectives via a shared MARBERTv2 encoder. Utilizing a hybrid BERT-CNN-BiLSTM-Attention architecture, they reported a competitive Macro F1-score of 0.9263, ranking **2nd** out of 15 participating teams.
- **CasbAI** (Username: *abdelazizchaima*): Participated in **Subtask 1**. They designed a Transformer-based approach leveraging MARBERT with a multi-seed ensemble strategy and Stochastic Weight Averaging (SWA). Through dialectal synonym replacement and Test-Time Augmentation (TTA), they reported achieving a Macro F1-score of 0.8462, ranking **3rd** among participating teams.
- **LinguArabic** (Username: *elham42*): Participated in **Subtask 1**. They fine-tuned MARBERT using a multi-stage preprocessing pipeline that included text normalization, dialect-aware lexical mapping, and confidence-based prediction adjustment. They reported achieving a Macro F1-score of 0.8333 on the official evaluation set (placing them **5th** closely on the leaderboard).
- **BDSI** (Username: *stdai*): Participated in **Subtask 1**. They developed a multi-model ensemble combining AraBERTv2 and CAMeLBERT with supervised contrastive learning. With a pipeline integrating dialect-aware preprocessing and rule-based post-processing, they reported a Macro F1-score of 0.83 (matching the **6th** placed system score of 0.8301).
- **WIT'INNOV & L3IA** (Username: *beatchalvador*): Participated in **Subtask 1**. They conducted a benchmarking study of five Transformer-based architectures, emphasizing a text normalization procedure to unify orthographic variations. They reported achieving an internal validation F1-score of 94.92%, while their official submission scored a Macro F1-score of 0.7596, placing them **13th** in the leaderboard.
- **Unnamed Team** (Username: *yumnahamdy*): Participated in **Subtask 2**. They investigated encoder-decoder and multilingual approaches for Arabic sentiment swap, leveraging an ensemble of AraBART and AraT5v2 with dialect-aware data augmentation. In their paper, they reported ranking **1st** in Subtask 2, achieving a BLEU score of 43.0, chrF of 65.36, and a sentiment preservation accuracy of 0.7554.
- **L3IA** (Username: *beatchalvador*): Participated in **Subtask 2**. They Addressed the task using a multi-step pipeline employing large language models (LLMs) to extract sentiment expressions, generate opposites, and reconstruct the sentence. They reported achieving 74.3% sentiment style accuracy, 27.22 BLEU, and 55.04 chrF, cleanly matching the **2nd** place leaderboard position.
- **VGU-M.Tech-AI** (Username: *asbichi362*): Participated in **Subtask 2**. They fine-tuned a multilingual T5 (mT5) model (ASBN-MT5) in a sequence-to-sequence manner. They reported conversion rates of 59.5% for positive to negative and 58.5% for negative to positive polarity swaps while achieving a BLEU score of 30.79 and a ChrF score of 54.08 on the official evaluation.

6. Results

Dialect Sentiment Analysis. Table 1 presents the results of the submissions for the Arabic Dialect Sentiment Analysis subtask, evaluated using the Macro F1-score. As shown in the table, the Musj1984 system achieved the highest F1-score of 0.9423, followed by TTLab with an F1-score of 0.9263. CasbAI ranked third, achieving an F1-score of 0.8462. Notably, the top-performing teams employed BERT-based variants as backbone encoder models. These models outperformed systems such as L3IA, which relied on LLMs for this task. This performance difference can be largely attributed to the Arabic training data used to train Arabic BERT variants, as well as the strong representation capabilities of BERT models, which make them well-suited for classification tasks. Additionally, systems that used ensemble approaches, such as Musj1984 and TTLab, significantly outperformed single-model approaches. This improvement can be attributed to the ability of ensemble methods to enhance prediction reliability by combining multiple model outputs, thereby increasing overall classification confidence.

Sentiment Swap. Table 2 presents the results for the Arabic Sentiment Swap subtask based on the Sentiment Style Accuracy, along with the generation quality metrics, BLEU, and chrF. As shown

#	Team/Participant Name	Macro F1-Score
1	Musj1984	0.9423
2	TTLab	0.9263
3	CasbAI	0.8462
4	University of Tripoli	0.8429
5	LinguArabic	0.8333
6	BDSI	0.8301
7	Mohammadnabulsi	0.8205
8	Gehadkamel	0.8205
9	Amani_Sh	0.8045
10	Tareqkhaled	0.7981
11	Mary-Sussex-ai	0.7981
12	Sarahmeid	0.7885
13	L3IA	0.7430
14	Amjad_Awad	0.6506
15	Astral_Fate	0.3686

Table 1: Results of the submitted systems for Sub-task 1: Arabic Dialect Sentiment Analysis.

in the table, the leaderboard results for this task show clear differences in how well systems balance sentiment transformation and content preservation. The Yumnamdy team ranks first with the highest sentiment style accuracy (0.7554), along with the best BLEU (43.00) and ChrF (65.36) scores. These results indicate a strong ability to change the sentiment while preserving the original meaning and structure. The L3IA team follows closely in second place with a similar sentiment accuracy (0.7430) but noticeably lower BLEU. These results indicate that the proposed model prioritizes sentiment transfer slightly more than exact wording. In contrast, the VGU-M.Tech-AI team performs poorly in sentiment accuracy (0.2368) despite having relatively better BLEU and ChrF than some competitors. This difference between the two metrics indicates that the model largely preserves the original text but does not effectively modify the sentiment. Overall, the results highlight that the key challenge of this task lies in achieving a balance between accurate sentiment transformation and maintaining the original semantic content.

7. Conclusion

This overview paper presented the AraSentEval shared task, which brought forward a set of challenges addressing sentiment classification for multiple Arabic dialects as well as the generative challenge of sentiment style transfer. By benchmarking on challenging tasks such as dialectal understanding and context-preserving sentiment swap, we aim to encourage future advances in both the analytical and generative capabilities of Arabic models. We hope that AraSentEval will be a basis for future research and encourage the development of more

dialect-aware and semantically consistent Arabic language models.

8. Bibliographical References

- Fahad J Abdu, Raed Mughaus, Shadi Abudalfa, Moataz Ahmed, and Ahmed Abdelali. 2025. An empirical evaluation of arabic text formality transfer: a comparative study. *Language Resources and Evaluation*, 59(4):4093–4153.
- Shadi I Abudalfa, Fahad J Abdu, and Maad M Alowaifeer. 2024. Arabic text formality modification: A review and future research directions. *IEEE Access*, 12:185117–185148.
- Maram I Alharbi, Salmane Chafik, Saad Ezzini, Ruslan Mitkov, Tharindu Ranasinghe, and Hansi Hettiarachchi. 2025a. Ahasis: Shared task on sentiment analysis for arabic dialects. In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 1–6.
- Maram I Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025b. Ador: Dataset for arabic dialects in hotel reviews: A human benchmark for sentiment analysis. In *Proceedings of the First Workshop on Advancing NLP for Low-Resource Languages*, pages 187–191.
- Maram I. Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025c. [Evaluating large language models on sentiment analysis in Arabic dialects](#). In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era*, pages 67–74, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.
- Ghadah Alwakid, Taha Osman, Mahmoud El Haj, Saad Alanazi, Mamoon Humayun, and Najm Us Sama. 2022. Muldasa: Multifactor lexical sentiment analysis of social-media content in non-standard arabic social media. *Applied Sciences*, 12(8):3806.
- Samhaa R. El-Beltagy, Mona El Kalamawy, and Abu Bakr Soliman. 2017. NileTMRG at SemEval-2017 task 4: Arabic sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 790–795.
- Mahmoud El-Haj. 2020. Habibi—a multi dialect multi national arabic song lyrics corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1318–1326.

#	Team/Participant Name	Sentiment Style Accuracy	BLEU	ChRF
1	Yumnamdy	0.7554	43.00	65.36
2	L3IA	0.7430	27.22	55.04
3	Astral_Fate	0.6950	20.33	45.51
4	VGU-M.Tech-AI	0.2368	30.79	54.08

Table 2: Results of the submitted systems for Subtask 2: Arabic Sentiment Swap.

Mo El-Haj. 2026. Tarab: A multi-dialect corpus of arabic lyrics and poetry. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026) at the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026)*, pages 37–46, Rabat, Morocco.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. [Delete, retrieve, generate: a simple approach to sentiment and style transfer](#). 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 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Raed Mughaus, Shadi Abudalfa, Hamzah Luqman, Fahad Abdu, Mohammed AlAli, Nawaf AlDowayan, and Ahmed Abdelali. 2026. Ma’aks: manually-curated parallel dataset for arabic text sentiment swap. *Language Resources and Evaluation*, 60(1):1.

Abdusalam F. Ahmad Nwesri, Nabila Almabrouk S. Shinbir, and Amani Bahlul Sharif. 2025. [Sentiment analysis on Arabic dialects: A multi-dialect benchmark](#). In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 86–91, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30.

TTLab at AraSentEval: SARF (صرف) Sentiment Analysis via Root-based Fusion for Multi-Dialectal Arabic

Ali Abusaleh, Bhuvanesh Verma, Prof. Dr. Alexander Mehler

Text Technology Lab (TTLab),
Goethe University Frankfurt
{a.abusaleh,verma,mehler}@em.uni-frankfurt.de

Abstract

Arabic sentiment analysis is challenged by morphological complexity and lexical variation across Arabic dialects, compounded by subjectivity in how speakers and writers express sentiment. In this paper, we present our submission for the AraSentEval 2026 Shared Task on Arabic Dialect Sentiment Analysis. We propose **SARF (صرف)** a multi-view architectural framework that integrates surface-level context with stemmed and rooted morphological perspectives using a shared MARBERTv2 encoder. Our system employs a hybrid BERT-CNN-BiLSTM-Attention architecture to capture both local sentiment n-grams and global sequential dependencies. Experimental results show that while individual morphological normalization strategies (stemming or rooting) may degrade performance, their joint integration via cross-morphological attention provides robust features across diverse dialects. Our final system achieved a competitive macro-F1-score of 0.9263, ranking 2nd out of 15 participating teams.

Keywords: NLP, Sentiment Analysis, Arabic analysis

1. Introduction

The advent of social media has fundamentally altered how individuals consume and produce information. Platforms such as X, Facebook, and Reddit are hubs for highly opinionated content that significantly influences public discourse. Similarly, online reviews on e-commerce websites provide personal insights and sentiments that drive consumer decision-making. In NLP, sentiment analysis is the task of polarity classification (where texts are typically labeled as positive, negative or neutral); sentiment analysis is treated as a core component of the broader area called opinion mining (Liu, 2012).

While sentiment analysis has been extensively researched across various languages, English remains the dominant focus due to its vast data availability (cf. Joshi et al. 2020). Despite being the fourth most used language on the web (Saloum et al., 2017), Arabic continues to trail behind English benchmarks in terms of state-of-the-art (SOTA) performance. A primary reason for this disparity is the prevalent use of informal Arabic dialects in online communication. While Modern Standard Arabic (MSA) is the formal register used in news, literature, and official documents, there are at least 22 distinct regional dialects used across the Arab world (Guellil and Azouaou, 2017). Because users interact on social media in their native dialects rather than MSA, Arabic text analysis presents distinct and substantial challenges for researchers.

To address these challenges, the Seventh Workshop on Open-Source Arabic Corpora and Pro-

cessing Tools (OSACT7) organized a shared task dedicated to Arabic Dialect Sentiment Analysis (Ezzini et al., 2026). This competition involves a multi-class classification challenge focused on identifying sentiment within texts written in various regional Arabic dialects. This task requires developing robust systems capable of navigating the substantial lexical and syntactic diversity found across the Arab world. To complete this task, we leveraged a state-of-the-art sentiment analysis framework based on the work of He and Abisado (2024). We extended this framework by introducing a novel architecture designed to incorporate morphological information specific to Arabic dialects (Procházka, 2021), thereby addressing their inherent linguistic complexity. Achieving a macro-F1-score of 0.9263, Our proposed system ranked 2nd out of 15 participating teams, demonstrating its effectiveness in handling dialectal variation.

2. Related work

While task-based sentiment analysis often struggles to generalize across domains or dialects (Shi and Agrawal, 2025), this challenge is particularly acute for Arabic, a highly context-sensitive language where meaning and sentiment can shift dramatically based on subsequent words or phrases. Bidirectional transformer models, such as BERT (Devlin et al., 2019), are naturally suited to this problem, as they capture context from both directions. This has been demonstrated in Arabic NLP through unitask and multitask models like HULMoNA (ElJundi et al., 2019) and ARABERT (Antoun et al., 2020). The practical ef-

fectiveness of such approaches is underscored by the 2021 Arabic Sentiment Competition at KAUST, where all top-ranking teams utilized variants of MARBERT (Abdul-Mageed et al., 2021; Alamro et al., 2021).

Pontiki et al. (2016) introduced the SemEval-2016 shared task on Aspect-Based Sentiment Analysis (ABSA). This task featured a specific track for Arabic hotel reviews, providing a dataset highly relevant to our task. Given the thematic overlap with our target domain, we leverage this corpus to enhance our model’s training phase.

He and Abisado (2024) proposed a hybrid framework incorporating BERT, CNN, and BiLSTM layers to mitigate the effects of polysemy and improve feature representation. In their architecture, the CNN component is utilized to capture local, spatial features within the text, while the BiLSTM layer extracts global, sequential dependencies. We adopt and extend this hybrid methodology to effectively navigate the morphological and syntactic diversity across the Arabic dialectal landscape.

3. Task Description and Dataset

The primary objective of this shared task on Arabic Dialect Sentiment Analysis is to identify sentiment, categorized as positive, negative, or neutral, in texts written in various Arabic dialects. This problem is framed as a supervised multi-class classification challenge. Formally, given an input sequence S representing a sentence in a specific Arabic dialect, the model is required to learn a mapping function $f : S \rightarrow Y$. The label space Y is defined as the set of sentiment categories:

$$Y = \{\text{positive, negative, neutral}\} \quad (1)$$

The goal is to predict the most probable label $\hat{y} \in Y$ for each input S , formulated as:

$$\hat{y} = \arg \max_{y \in Y} P(y | S; \theta) \quad (2)$$

where θ represents the parameters of the classification model.

To facilitate the development of a robust classification model, the shared task utilizes the **Multi-Dialect-Sent (MDS-3)** dataset. This corpus consists of 3,000 sentences (1,731 in train set) annotated across three sentiment labels and covers four major regional dialects: Saudi (SA), Jordanian (JO), Moroccan (MA), and Egyptian (EG) (Alharbi et al., 2025c,a,b). The dataset was originally sourced from hotel reviews and subsequently translated and verified by native speakers of each respective dialect to ensure linguistic authenticity.

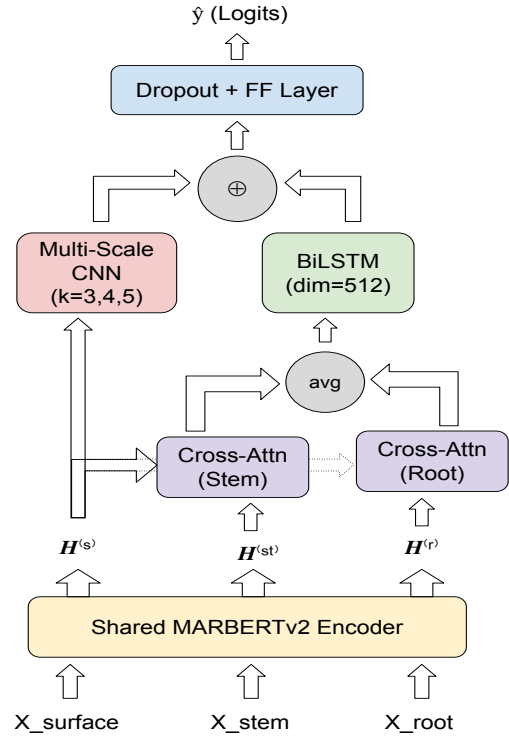


Figure 1: **SARF (صرف)** Architecture: A multi-view pipeline with shared MARBERTv2 encoder and CNN/BiLSTM dual-path.

4. Methodology

To develop a robust sentiment analysis system capable of addressing the dialectal nuances of Arabic, our approach integrates an established state-of-the-art (SOTA) framework with a novel architectural extension. This design specifically targets the inherent morphological and linguistic complexities present across diverse Arabic dialects. In the following subsections, we provide a detailed description of these primary components and our overall modeling strategy.

4.1. Cross morphological attention

We propose a multi-view architectural framework that captures linguistic nuances by representing a single input sentence through three parallel perspectives: the surface form, the stemmed form, and the rooted form. This approach ensures that the model maintains a balance between raw contextual semantics and underlying morphological structures.

4.1.1. Multi-View Encoding

Each linguistic view is first tokenized and subsequently mapped to a high-dimensional space using a shared pretrained transformer encoder $E(\cdot)$.

Category	Precision	Recall	F1-score
Dialect-Specific Ensemble System			
Darija	0.902	0.902	0.902
Egyptian	0.889	0.889	0.889
Jordanian	0.970	0.970	0.970
Saudi	0.967	0.967	0.967
Overall			
Ensemble System	0.9317	0.9317	0.9316
BERT-CNN-BiLSTM	0.9429	0.9421	0.9425
BERT-CNN-BiLSTM + Cross Morph. Attn	0.9398	0.9378	0.9388

Table 1: Performance of the Ensemble and BERT-CNN-BiLSTM-Self-Attention system. Dialect Specific results are using Ensemble system only.

Formally, let the input sequences be denoted as $X^{(s)}$, $X^{(st)}$, and $X^{(r)}$ for the surface, stemmed, and rooted views, respectively. The encoder generates contextualized representations as follows:

$$H^{(s)} = E(X^{(s)}), H^{(st)} = E(X^{(st)}), H^{(r)} = E(X^{(r)}) \quad (3)$$

where each representation $H^{(\cdot)} \in \mathbb{R}^{T \times d}$ corresponds to a sequence of length T with hidden dimension d .

4.1.2. Morphological Interaction and Fusion

To capture the dependencies between surface-level semantics and morphological abstractions, we employ a cross-attention mechanism. Here, the surface representation $H^{(s)}$ serves as the query, while the stemmed and rooted representations act as key-value pairs. This allows the model to align inflectional variations with their base forms:

$$\tilde{H}^{(st)} = \text{Attn}(H^{(s)}, H^{(st)}, H^{(st)}) \quad (4)$$

$$\tilde{H}^{(r)} = \text{Attn}(H^{(s)}, H^{(r)}, H^{(r)}) \quad (5)$$

The resulting features are then integrated into a unified representation $H^{(f)}$ through an element-wise averaging operation:

$$H^{(f)} = \frac{1}{3} (H^{(s)} + \tilde{H}^{(st)} + \tilde{H}^{(r)}) \quad (6)$$

4.2. BERT-CNN-BiLSTM

Building upon the framework of [He and Abisado \(2024\)](#), our model processes the surface level features $H^{(s)}$ and the morphologically enriched representation $H^{(f)}$ through parallel feature extraction pathways to capture both local and global linguistic patterns.

4.2.1. Feature Extraction Pathways

The local features, H_{local} , are extracted using a Convolutional Neural Network (CNN). For a filter W_c and bias b_c , the operation is defined as:

$$H_{local} = \text{ReLU}(W_c * H^{(s)} + b_c) \quad (7)$$

Simultaneously, global sequential dependencies are captured using a Bidirectional LSTM (BiLSTM). The representation H_{global} is the concatenation of the forward ($\vec{h}$) and backward ($\overleftarrow{h}$) hidden states:

$$H_{global} = [\overrightarrow{\text{LSTM}}(H^{(f)}) \oplus \overleftarrow{\text{LSTM}}(H^{(f)})] \quad (8)$$

4.2.2. Feature Fusion and Classification

The local and global features are subsequently fused via concatenation to form a comprehensive representation, $H_{total} = [H_{local} \oplus H_{global}]$. This composite vector is passed through a fully connected layer to compute the final sentiment distribution:

$$\hat{y} = \text{Softmax}(W_{fc}H_{total} + b_{fc}) \quad (9)$$

where W_{fc} and b_{fc} are the learnable weights and bias, respectively. The resulting output $\hat{y}$ represents the probability distribution over the label set $Y = \{\text{positive, negative, neutral}\}$, as illustrated in the full pipeline in [Figure 1](#).

5. Experimental Setup

We utilize the SemEval-2016 Arabic reviews dataset, pre-training for 10 epochs and fine-tuning for 7 at a reduced learning rate (1.24×10^{-5}). Data preprocessing is performed using [PyArabic \(Zerrouki, 2010\)](#), [Tashaphyne \(Zerrouki, 2012b\)](#), and [Qalsadi \(Zerrouki, 2020, 2012a\)](#) to normalize the text, and extracting surface, stem, and root features. Our architecture employs a 128-dimensional LSTM and a 200-filter CNN (kernels: 3, 4, 5) with 0.3 dropout, 0.02 weight decay, and a batch size of 128 (8 for testing). The model is optimized using AdamW with an initial learning rate of 1.24×10^{-4} . Implementation details and the cleaning pipeline are available at [GitHub](#).

6. Results and Discussion

6.1. Performance Overview

We evaluated **SARF** (صرف) against several competitive baselines, including an ensemble of SOTA Arabic transformers (MARBERTv2, AraBERT, and QARIB). As shown in Table 1, our hybrid **BERT-CNN-BiLSTM** approach achieved the highest performance on the AraSentEval DEVELOPMENT set with a **macro-F1-score of 0.9425**, outperforming the ensemble system (0.9316).

6.2. Synergy of Morphological Views

Our normalization results (Table 4) confirm that neither stemming nor rooting is a *silver bullet* in isolation. Rooting alone tends to remove sentiment-critical lexical nuances. However, the multi-view approach of **SARF** (صرف)—which aligns these views via cross-attention—successfully captures the stability of the root without losing the sentiment of the surface form. This confirms that the interaction between different morphological granularities is essential for handling the noise inherent in dialectal Arabic.

6.3. Final Shared Task Results

The final performance of our system on the “blind” test set provided by the AraSentEval 2026 organizers is reported in Table 2.

Metric	Value
Macro Precision	0.9262
Macro Recall	0.9262
Macro-F1 Score	0.9262
Balanced Accuracy	0.8941

Table 2: Final results on the AraSentEval 2026 Shared Task test set.

7. Error Analysis

A consistent observation across all tested models and strategies was the significantly lower performance on the *Neutral* label. To investigate the root cause of this trend, we conducted a two-fold analysis: 1) a human re-annotation study of a sample training data and 2) an examination of model predictions. The methodology for the re-annotation is detailed in Appendix A.

7.1. Human Re-annotation and Label Noise

Our first analysis focused on assessing the quality of the original ground-truth labels through two

distinct settings: **Relaxed Setting**: We analyzed all samples re-annotated by our human experts, regardless of the number of annotators per sample. Comparing these to the original training labels, we calculated a moderate Cohen’s Kappa (κ) of 0.5360. Notably, of the 60 instances labeled as *Neutral* in the original dataset, our annotators re-classified 11 as *Negative* and 23 as *Positive*.

Strict Setting: We restricted the analysis to 68 samples where consensus was reached by more than one annotator. In this setting, the inter-rater agreement with the original labels dropped significantly to a low $\kappa = 0.2958$. Among the 29 original *Neutral* samples in this subset, 15 were re-identified as *Positive* and 6 as *Negative*.

Sample	Original Label	Annotated Label
خدمة مزيانة، فندق ف موقع مزيان قريب من المطار المحلي و الدولي في أندهري شرق مومباي، حنا عندنا Good service, hotel in a good location near the local and international airport in Andheri East, Mumbai, we have...	Neutral	Positive
المساحة كانت مزيانة و كان فيها حمام سباحة صغير و مشدود و فيه شلال فوسطها The swimming was good; there was a small, well-maintained pool with a waterfall in the middle	Neutral	Positive
الخدمة مزيانة، الأكل مزيان، الغرف مريحة، وخلال فترة وجودنا، شافنا فرح يقوم بالفندق، و كانو The service was good, the food was good, the rooms were comfortable, and during our stay, we saw a wedding at the hotel	Neutral	Positive

Figure 2: Re-annotation of *Hidden Positives*. These examples demonstrate how dialect-specific positive sentiment (such as the use of مزيانة for good) was originally overlooked and labeled as ‘neutral’, but accurately captured in our new annotations.

7.2. Model Prediction

Our error analysis suggests that the model internalized subjective, dialect-agnostic biases in the shared-task annotations. On instances where our re-annotations disputed the original labels, the model memorized the incorrect human bias 46.9% of the time, though its underlying pre-trained representations demonstrated some robustness by generalizing correctly in 40.6% of these cases. Crucially, this inherited noise disproportionately penalized specific regional dialects: while the model achieved moderate accuracy on the Jordanian (66.7%) and Saudi (64.7%) subsets, its performance degraded sharply to 44.4% on Darija. This degradation is directly linked to an inherited *safe bias* from the original annotators, where the model most frequently misclassified both positive and negative instances as Neutral (accounting for 86% and 100% of their respective errors).

Figure 2 shows the original and new annotations for a few samples. This discrepancy indicates that the model’s suboptimal performance on neutral text is likely due to false signals and label

noise within the training data, rather than architectural limitations.

8. Conclusion

In this paper, we described our submission to the AraSentEval 2026 Shared Task on Arabic Dialect Sentiment Analysis. Our system, **SARF** (صرف), addresses the morphological complexity of 4 regional dialects by integrating surface-level context with stemmed and rooted views via a cross-attention mechanism. By building a hybrid BERT-CNN-BiLSTM-Attention framework on top of shared MARBERTv2 encoder, we successfully captured both local sentiment n-grams and global sequential dependencies. Our results demonstrate that while individual morphological normalization (rooting) can be reductive, a multi-view approach provides a robust representation that is invariant to dialectal noise. Furthermore, our error analysis revealed significant label noise in the *Neutral* class, suggesting that the current performance ceiling in Arabic Dialect Sentiment Analysis may be limited more by annotation consistency than by model architecture. Our system ranked 2nd, indicating that morphologically aware hybrid models can be effective for low-resource, high-variance dialectal tasks.

9. Limitations & Ethical Considerations

Limitations Our system faces three primary limitations. First, the **SARF** (صرف) architecture requires three parallel encoding passes (surface, stem, and root), which triples the computational overhead during inference compared to standard Transformer models. This may limit scalability for massive real-time streams. Second, our reliance on the `Qalsadi` and `Tashaphyne` tools for rooting means that errors in the rule-based morphological analyzer can propagate through the cross-attention layers. Finally, while we investigated four major dialects, the model’s performance on more localized or “deep” dialectal expressions (especially in the Maghrebi cluster) remains lower than its performance on Levantine and Gulf dialects, likely due to the linguistic distance from the MARBERTv2 pre-training corpus.

Ethical Considerations Our error analysis specifically identified a “safe bias” in the original annotations that disproportionately mislabeled Darija (Moroccan) expressions. We emphasize that the systemic misclassification of specific regional dialects can lead to linguistic exclusion in AI-driven services. To mitigate this, we advocate

for the inclusion of native speakers from across the Mashreq (المشرق) and Maghreb (المغرب) in future annotation efforts. Our code and fine-tuned weights are released for academic use to promote transparency in dialectal NLP research.

References

- Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. [ARBERT & MARBERT: Deep bidirectional transformers for Arabic](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.
- Hind Alamro, Manal Alshehri, Basma Alharbi, Zuhair Khayyat, Manal Kalkatawi, Inji Ibrahim Jaber, and Xiangliang Zhang. 2021. [Overview of the arabic sentiment analysis 2021 competition at kaust](#).
- Maram Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025a. Evaluating large language models on arabic dialect sentiment analysis. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing (RANLP)*.
- Maram I Alharbi, Salmane Chafik, Saad Ezzini, Ruslan Mitkov, Tharindu Ranasinghe, and Hansi Hettiarachchi. 2025b. Ahasis: Shared task on sentiment analysis for arabic dialects. In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 1–6.
- Maram I Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025c. Ador: Dataset for arabic dialects in hotel reviews: A human benchmark for sentiment analysis. In *Proceedings of the First Workshop on Advancing NLP for Low-Resource Languages*, pages 187–191.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of](#)

- deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Obeida ElJundi, Wissam Antoun, Nour El Droubi, Hazem Hajj, Wassim El-Hajj, and Khaled Shaaban. 2019. [hULMonA: The universal language model in Arabic](#). In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 68–77, Florence, Italy. Association for Computational Linguistics.
- Saad Ezzini, Shadi Abudalfa, Maram Alharbi, Salmane Chafik, Hamzah Luqman, Mo El-Haj, Paul Rayson, Samaher Alghamdi, and Reem Alotaibi. 2026. AraSentEval: A shared task on sentiment analysis and swapping in arabic. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with the Language Resources and Evaluation Conference (LREC 2026)*, Palma, Mallorca, Spain.
- Imène Guellil and Faïçal Azouaou. 2017. [Asda: Analyseur syntaxique du dialecte alg {\'e} rien dans un but d’analyse s {\'e} mantique](#). *ArXiv preprint*, abs/1707.08998.
- Aixiang He and Mideth Abisado. 2024. Text sentiment analysis of douban film short comments based on bert-cnn-bilstm-att model. *Ieee Access*, 12:45229–45237.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. [The state and fate of linguistic diversity and inclusion in the NLP world](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.
- Bing Liu. 2012. *Sentiment analysis and opinion mining*, volume 5 of *Synthesis lectures on human language technologies*. Morgan & Claypool Publishers.
- Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.
- Stephan Procházka. 2021. *Arabic Dialectology*, Cambridge Handbooks in Language and Linguistics, page 214–243. Cambridge University Press.
- Said A Salloum, Mostafa Al-Emran, Azza Abdel Monem, and Khaled Shaalan. 2017. Using text mining techniques for extracting information from research articles. In *Intelligent natural language processing: Trends and Applications*, pages 373–397. Springer.
- Zhiqiang Shi and Ruchit Agrawal. 2025. [A comprehensive survey of contemporary Arabic sentiment analysis: Methods, challenges, and future directions](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3760–3772, Albuquerque, New Mexico. Association for Computational Linguistics.
- Taha Zerrouki. 2010. [pyarabic, an arabic language library for python](#).
- Taha Zerrouki. 2012a. [qalsadi, arabic mophological analyzer library for python](#).
- Taha Zerrouki. 2012b. [Tashaphyne, arabic light stemmer](#).
- Taha Zerrouki. 2020. Towards an open platform for arabic language processing.

A. Annotation Methodology

To validate the quality of the original labels, we randomly sampled 30 instances per dialect from the training set. The re-annotation task was performed by native speakers of the respective dialects, all of whom were university students or graduates. The annotator pool ranged in age from 22 to 50 years, with a mean age of 29.

Dialect	Pos	Neg	Neu
Moroccan (Darija)	17	5	5
Saudi	26	21	18
Jordanian	45	45	28

Table 3: Distribution of sentiment labels across the re-annotated dialects.

A.1. Annotator Distribution and Data Collection

The distribution of annotators was as follows: 11 for Jordanian, 5 for Saudi, 5 for Moroccan (Darija), and 2 for Egyptian. Due to a lack of completed responses from the Egyptian cohort, this dialect was excluded from the subsequent analysis. Following the collection phase, the final re-annotated dataset comprised 119 instances for Jordanian, 68 for Saudi, and 30 for Moroccan.

A.2. Consensus and Analysis Settings

To determine the final labels for our study, we applied a majority voting scheme to each instance. We then categorized the data into two analytical settings:

- **Strict Analysis:** Only instances with two or more annotations were considered to ensure higher label confidence.
- **Relaxed Analysis:** All instances with at least one expert annotation were included.

The resulting label distributions and a comparison with the original task labels are summarized in Table 3.

B. Language-specific Normalization

As shown in Table 4, across both models, applying stemming or rooting independently does not yield consistent improvements over the baseline without normalization. In particular, rooting alone leads to a noticeable degradation in performance, indicating that collapsing words to their morphological roots removes sentiment relevant lexical distinctions. Similarly, stemming alone shows marginal differences relative to the baseline, suggesting limited added value in the presence of strong pre-trained representations. In contrast, combining stemming and rooting consistently improves performance for both MARBERT and MARBERTv2. These findings suggest that multi-granularity morphological normalization allows the model to benefit from **dialect-level grouping via roots** while preserving surface-level and contextual information necessary for sentiment discrimination.

Setup	Macro-F1	Accuracy	Macro Precision	Macro Recall
MARBERT				
No stem, no root	0.8658	0.8719	0.8692	0.8656
Stem only	0.8627	0.8686	0.8645	0.8626
Root only	0.8522	0.8587	0.8544	0.8527
Stem + root	0.8762	0.8818	0.8766	0.8770
MARBERTv2				
No stem, no root	0.8838	0.8884	0.8840	0.8850
Stem only	0.8822	0.8876	0.8849	0.8819
Root only	0.8813	0.8868	0.8822	0.8815
Stem + root	0.8951	0.9000	0.8959	0.8951

Table 4: Cross-validated performance on AraSentEval **Validation-dataset** under different normalization strategies.

MYM at AraSentEval: A Comparative Study of Arabic Sentiment Swap Models

Yumna Hamdy, Mohab EIDamhougy, Yomna Eid, Ensaf Hussien

Nile University, Egypt

Cairo, Egypt

Y.Hamdy2457@nu.edu.eg, M.Tarek2317@nu.edu.eg, YEid@nu.edu.eg, EnMohamed@nu.edu.eg

Abstract

Sentiment swap is a controlled text generation task that rewrites a sentence by inverting its sentiment polarity while preserving semantic content and fluency. In this paper, we present our system for AraSentEval 2026 Subtask 2 on Arabic sentiment swap, a particularly challenging problem due to Arabic's rich morphology and dialectal variation. We investigate multiple modeling paradigms, including encoder–decoder and multilingual approaches, and propose an enhanced system that combines targeted data augmentation and ensemble learning. Specifically, we augment underrepresented dialectal patterns to improve robustness and ensemble two Arabic-focused sequence-to-sequence models, AraBART and AraT5v2. Experiments are conducted on the MA'aks parallel dataset under fine-tuned settings.

Our system ranked first in AraSentEval 2026 Subtask 2, achieving a BLEU score of 43.0, chrF of 65.36, and sentiment preservation accuracy of 0.7554. The results demonstrate that dialect-aware augmentation together with model ensembling substantially improves sentiment-controlled generation in Arabic and establishes strong baselines for future research in low-resource sentiment manipulation.

Keywords: Arabic NLP, sentiment swap, style transfer, AraSentEval, text generation

1. Introduction

Sentiment analysis is traditionally formulated as a classification task that assigns polarity labels to text. Recent advances in neural generation have enabled sentiment swap, where a sentence is rewritten to express the opposite sentiment while preserving meaning and fluency. This task is substantially more challenging than classification because models must accurately identify and modify sentiment-bearing expressions.

Arabic sentiment swap is particularly difficult due to rich morphology, dialectal variation, and frequent implicit sentiment cues. While Arabic sentiment classification has been widely studied, sentiment-controlled generation remains underexplored. The MA'aks dataset and the AraSentEval shared task provide a standardized benchmark for this problem.

In this work, we present our system for AraSentEval 2026 Subtask 2. We combine dialect-aware data augmentation with an ensemble of two Arabic encoder–decoder models, AraBART and AraT5v2. Our approach ranked first in the shared task, demonstrating the effectiveness of augmentation and model ensembling for Arabic sentiment swap.

Contributions: (i) a top-performing AraBART–AraT5v2 ensemble, (ii) dialect-aware augmentation, and (iii) evaluation of Arabic and multilingual models.¹

¹The code

2. Related Work

Sentiment style transfer has drawn growing interest in controlled text generation, but Arabic remains under-resourced. Mughaus et al. (Mughaus et al., 2026) introduced MA'AKS, the first manually curated parallel dataset for Arabic sentiment swap, with about 5,000 Modern Standard Arabic sentence pairs of opposite polarity.

Using MA'AKS, they evaluated AceGPT, JAIS, and Llama-3 under zero-shot, few-shot, and fine-tuning settings. Llama-3-8B-Instruct achieved the best zero-shot semantic preservation (BLEU 43.68 PTN / 44.9 NTP), while supervised fine-tuning produced the strongest overall gains, reaching 59.9 PTN / 57.1 NTP (Mughaus et al., 2026).

However, sentiment control remains difficult, especially for dialectal Arabic, where all models show notable performance drops. The AraSentEval shared task (Ezzini et al., 2026) extended evaluation to a larger benchmark, but systematic study of encoder–decoder architectures and full fine-tuning remains limited.

In this work, we address these gaps by evaluating Arabic-specialized encoder–decoder models (AraT5v2 and AraBART), introducing dialect-aware data augmentation, and exploring ensembling to improve sentiment control and semantic preservation.

3. Task and Dataset

3.1. Task Definition

Sentiment swap is a controlled generation task that reverses the sentiment of an input sentence while preserving meaning and fluency. Given a sentence x with label $s \in \{positive, negative\}$, the goal is to generate y expressing the opposite sentiment $\bar{s}$. This task is challenging in Arabic due to rich morphology and implicit sentiment cues.

3.2. MA'aks Dataset

We evaluate on the MA'aks dataset, a parallel corpus for Arabic sentiment swap with separate training, validation, and test files covering MSA, Gulf, Levantine, Egyptian, and Moroccan dialects. The training set contains 6,262 pairs (54.58% positive, 45.42% negative), and the validation set includes 1,315 pairs (61.9% positive, 38.1% negative). The test set comprises 647 unlabeled sentences. Each labeled instance pairs a source sentence with a sentiment-inverted rewrite.

4. Methodology

We investigate Arabic sentiment swap using multiple modeling paradigms and propose an ensemble-based system. Our study compares encoder-decoder and decoder-only approaches, analyzes fine-tuning effects, explores dialect-aware augmentation, and introduces a CAMELBERT-based selection mechanism.

4.1. Arabic Encoder-Decoder Models

We focus on encoder-decoder architectures, which naturally support controlled rewriting. We evaluate two Arabic-focused models: AraT5v2 and AraBART.

We fine-tune AraBART on MA'aks, obtaining substantial gains. AraT5v2 is directly fine-tuned and consistently performs slightly better, confirming the effectiveness of Arabic-pretrained sequence-to-sequence models.

4.2. Multilingual Encoder-Decoder Model

We also evaluate MT5, a multilingual encoder-decoder model. Despite broad language coverage, MT5 performs poorly on Arabic sentiment swap and incurs higher computational cost, indicating that multilingual pretraining alone is insufficient for fine-grained sentiment manipulation.

4.3. Decoder-Only Model

For comparison, we fine-tune the decoder-only GPT-J-6B model. Although it generates fluent text, it struggles to reliably invert sentiment and consistently underperforms relative to encoder-decoder models, reinforcing the suitability of sequence-to-sequence architectures.

4.4. Dialect-Aware Data Augmentation

To analyze dialect coverage and explore robustness, We used the MARBERTv2 Arabic Written Dialect Classifier to label dialects in the training data. The original distribution over 6,263 instances was skewed toward MSA (Modern Standard Arabic): MSA (49.77%), GLF (19.99%), EGY (13.09%), LEV (10.49%), and MAGHREB (6.66%).

To improve balance, we performed targeted augmentation on underrepresented dialects (EGY, LEV, MAGHREB). The dataset was split into minority and majority subsets, and lightweight label-preserving transformations were applied to the minority samples, including tatweel removal, controlled elongation, punctuation variation, and optional emoji insertion. One augmented instance was generated per selected sample while keeping the target text unchanged.

The augmented dataset grew to 8,157 instances with a more balanced distribution: MSA (38.22%), EGY (20.11%), LEV (16.11%), GLF (15.35%), and MAGHREB (10.21%). However, this strategy did not yield consistent gains.

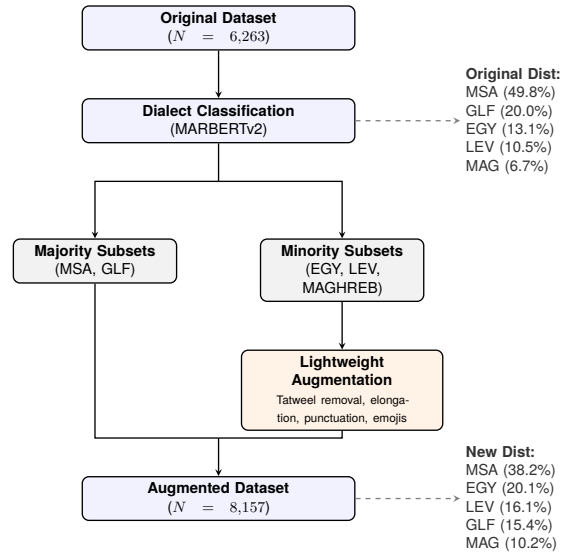


Figure 1: Overview of the dialect-aware data augmentation pipeline. The original dataset is classified and split, applying targeted lightweight transformations only to underrepresented dialects to achieve a more balanced distribution.

4.5. Training Strategy

Encoder–decoder models are fine-tuned using standard sequence-to-sequence cross-entropy loss, enabling explicit learning of sentiment-inversion mappings. Fine-tuned Arabic models provide the strongest baseline performance.

4.6. Ensemble Inference

Our final submission ensembles AraT5v2 and AraBART. Both models are trained independently, and during inference each input sentence is processed by both models to produce candidate rewrites.

CAMeLBERT-DA is then used as a sentiment judge, selecting the candidate whose predicted polarity best matches the target. This selection-based ensemble improves sentiment correctness while maintaining fluency.

5. Results and Discussion

We evaluate generation quality using BLEU, chrF, Loss and Accuracy. BLEU measures n-gram precision between generated and reference sentences, reflecting lexical fidelity. Together, these metrics provide a complementary assessment of semantic consistency and surface-level accuracy in the sentiment swap task.

To explicitly verify polarity inversion, we additionally employ CAMeLBERT-DA, a dialect-aware BERT-based Arabic sentiment classifier. The model encodes sentences using bidirectional self-attention and predicts sentiment from contextualized representations, providing a sentiment-oriented evaluation signal that complements reference-based metrics.

Table 1 shows that fine-tuned Arabic encoder–decoder models substantially outperform multilingual and decoder-only approaches for Arabic sentiment swap. Overall, models pretrained specifically for Arabic demonstrate stronger sentiment inversion and better semantic preservation, confirming the importance of language-specific pretraining for controlled generation in morphologically rich languages such as Arabic.

Model	Loss	BLEU	chrF	Acc.
AraT5v2	0.99	42.71	65.0	74.61
AraBART	0.82	42.82	65.53	69.66
mT5	0.998	54.44	40.02	53.5
GPT-J-6B	0.32	53	33.7	44

Table 1: Performance comparison for each model individually on the MA’aks test set.

5.1. Encoder–Decoder vs. Decoder-Only Models

Our experiments show a clear advantage of encoder–decoder architectures over decoder-only generation. While GPT-J-6B produces fluent outputs, it struggles to reliably invert sentiment and preserve meaning, resulting in weaker overall performance. In contrast, sequence-to-sequence models benefit from explicit input–output alignment, making them better suited for controlled sentiment rewriting.

5.1.1. Multilingual vs. Arabic Encoder–Decoder Models

Within encoder–decoder approaches, Arabic-focused models consistently outperform the multilingual mT5. Despite its broad pretraining, mT5 fails to capture fine-grained sentiment transformations in Arabic and exhibits substantially weaker chrF and accuracy scores. These results indicate that multilingual pretraining alone is insufficient for sentiment-controlled generation in morphologically rich languages.

In contrast, AraT5v2 and AraBART demonstrate strong and stable performance across metrics. Their Arabic-specific pretraining enables more precise manipulation of sentiment-bearing expressions. Given these findings, our subsequent experiments and system design focus primarily on Arabic-pretrained models rather than multilingual alternatives.

5.2. Ensemble Performance

We further evaluate the proposed ensemble strategy combining AraT5v2 and AraBART. When used individually under fine-tuned settings, AraT5v2 achieves BLEU 42.71, chrF 65.0, and sentiment preservation 0.7461, while AraBART obtains BLEU 42.82, chrF 65.53, and sentiment preservation 0.6966. Although AraBART slightly outperforms AraT5v2 in lexical metrics, AraT5v2 provides stronger sentiment preservation.

By leveraging CAMeLBERT-DA as a selection mechanism during inference, the ensemble effectively combines the complementary strengths of both models. This results in our best-performing system, achieving BLEU 43.0, chrF 65.36, and sentiment preservation 0.7554, which secured first place in AraSentEval 2026 Subtask 2. These findings confirm that selection-based ensembling improves sentiment correctness while maintaining high generation quality.

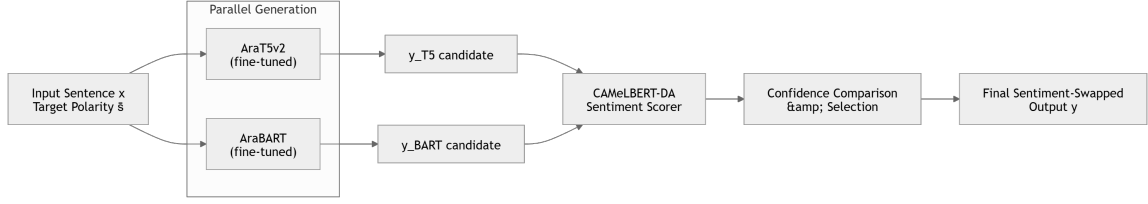


Figure 2: Ensemble Parallel AraT5v2/AraBART generation with CAMELBERT-DA-based selection.

6. Error Analysis

6.1. Major Failure Modes

Despite strong overall performance, several systematic failure patterns remain. The most prominent source of error is dialect imbalance. Although we experimented with targeted augmentation, the approach did not fully resolve skew in dialect coverage. Performance degradation is most visible on underrepresented varieties and on sentences exhibiting dialectal ambiguity, particularly between MSA and Gulf Arabic, where lexical overlap complicates reliable polarity manipulation.

The model also struggles with informal phenomena such as emojis and idiomatic expressions, where sentiment is conveyed implicitly rather than through explicit polarity markers. In such cases, the system often preserves fluency but fails to correctly invert sentiment.

6.2. Qualitative Examples

Representative challenging cases include:

- Arabic (original):** مر رهيب ورائع وطعم اللذة يستحق مليون نجمة
Transliteration: mar rahīb wa rā'ī° wa ṭa°m al-ladhdha yastahīq milyūn najma.
Translation: 'It is amazing and wonderful, and the delicious taste deserves a million stars.'
- Arabic (original):** تطبيق رائع والطلب منه كثير سهل
Transliteration: taṭbīq rā'ī° wa al-ṭalab minhu katīr sahl.
Translation: 'A wonderful app, and ordering from it is very easy.'

These examples illustrate failures involving implicit sentiment cues, emoji-driven polarity, and dialect-sensitive phrasing.

6.3. Root Cause Analysis

The primary limitation stems from dialect distribution mismatch and cross-dialect similarity. When dialect boundaries are blurred (e.g., MSA vs. Gulf), the system may misidentify sentiment-bearing

spans or apply inappropriate rewrites. Our augmentation strategy improved dialect balance statistically but did not sufficiently enhance model robustness to these ambiguities.

6.4. Future Work

Future work can extend this study in several directions. First, more advanced fine-tuning strategies such as multi-stage fine-tuning and curriculum learning could gradually expose models to increasingly complex sentiment transformations. Second, stronger augmentation methods such as dialect-controlled paraphrasing and back-translation may better address dialect imbalance. Third, synthetic data generation using large language models could help create larger distribution-matched corpora for robust fine-tuning. Finally, alternative objectives beyond standard cross-entropy, including sentiment-aware or contrastive losses, may better enforce polarity correctness while preserving semantic content.

7. Conclusion

In this paper, we study Arabic sentiment swap as a controlled generation task requiring precise polarity inversion while preserving meaning and fluency. Experiments on the MA'AKS dataset show that reliable sentiment manipulation in Arabic benefits more from language-specific supervised learning than from purely multilingual approaches.

Our results demonstrate that Arabic encoder-decoder models, particularly AraT5v2 and AraBART, outperform multilingual and decoder-only baselines. We further propose a selection-based ensemble that leverages their complementary strengths using CAMELBERT-DA as a sentiment-aware judge. The proposed system achieves state-of-the-art performance and ranks first in AraSentEval 2026 Subtask 2.

Error analysis reveals persistent challenges, including dialect imbalance, implicit sentiment cues, and emoji-driven expressions. Overall, this work advances sentiment-controlled generation in Arabic and establishes strong baselines for future research on style transfer in morphologically rich and dialectally diverse languages.

8. Bibliographical References

- M. Elmadany, A. Elbehery, and M. Abdelrahman. 2021. Arabart: Transformer-based model for arabic text generation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.
- Saad Ezzini, Shadi Abudalfa, Maram Alharbi, Salmane Chafik, Hamzah Luqman, Mo El-Haj, Paul Rayson, Samaher Alghamdi, and Reem Alotaibi. 2026. AraSentEval: A shared task on sentiment analysis and swapping in arabic. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with the Language Resources and Evaluation Conference (LREC 2026)*, Palma, Mallorca, Spain.
- A. Inoue, M. Alsharif, and H. Mubarak. 2021. Camelbert: A collection of pretrained transformer models for arabic. In *Proceedings of the ACL Workshop on NLP for Arabic*.
- A. Nagoudi, M. El-Beltagy, and A. Abdelali. 2022. Arat5: Text-to-text transformers for arabic language generation. In *Proceedings of the ACL Workshop on NLP for Arabic*.
- B. Wang and A. Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model. Technical report, EleutherAI.
- L. Xue, N. Constant, A. Roberts, et al. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of NAACL*.

9. Language Resource References

- Raed Mughaus, Shadi Abudalfa, Hamzah Luqman, Fahad Abdu, Mohammed AlAli, Nawaf AlDowayan, and Ahmed Abdelali. 2026. Ma'aks: manually-curated parallel dataset for arabic text sentiment swap. *Language Resources and Evaluation*, 60(1):1.

L3IA at AraSentEval Shared Task: LLM-Based Multi-Step Pipeline for Arabic Sentiment Swap

Hamza Alami, Mohamed M'haouach, Kaouthar Elyoussoufi, Abdessamad Benlahbib

L3IA Laboratory, Faculty of Sciences Dhar El Mahraz,
Sidi Mohamed Ben Abdellah University, Fez, 30003, Morocco
{hamza.alami5, mohamed.mhaouach, kaouthar.elyoussoufi,
abdessamad.benlahbib}@usmba.ac.ma

Abstract

This paper describes our system submitted to the AraSentEval 2026 Shared Task, Subtask 2: Arabic Sentiment Swap. The task requires rewriting Arabic sentences to invert their sentiment polarity while preserving the core meaning. We propose a multi-step pipeline approach that uses large language models (LLMs). Our method decomposes the sentiment inversion problem into three stages: (1) sentiment expression extraction, where the model identifies all sentiment-bearing words and phrases in the input sentence; (2) opposite expression generation, where each identified expression is replaced by its semantic opposite; and (3) sentence reconstruction, where the final output is assembled to ensure grammatical correctness and natural fluency. Our system achieves 74.3% sentiment style accuracy, 27.22 BLEU, and 55.04 chrF on the official test set.

Keywords: Sentiment Analysis, Arabic sentiment swap, NLP

1. Introduction

Sentiment analysis is a well-studied area in natural language processing (NLP). However, the reverse task — transforming a sentence's sentiment from positive to negative or vice versa — remains a challenging and less explored problem. This task, known as sentiment style transfer or sentiment swap, requires models to go beyond classification and actually generate text with the opposite polarity while keeping the original meaning intact (Jin et al., 2022; Hu et al., 2022; Toshevskaya and Gievska, 2022).

Arabic presents additional challenges for this task. It is a morphologically rich language with a complex system of roots, patterns, and affixes. Moreover, the Arab world exhibits significant dialectal variation, with Egyptian, Gulf, Levantine, and Maghrebi dialects differing from Modern Standard Arabic (MSA) in vocabulary, grammar, and usage (Younes et al., 2020; Joshi et al., 2025). Social media content, which forms the basis of the dataset used in this shared task, is especially challenging because it mixes dialects, uses informal spelling, and includes emojis and code-switching (Hajbi et al., 2022).

The AraSentEval 2026 Shared Task (Ezzini et al., 2026) addresses these challenges through two subtasks. Subtask 1 focuses on sentiment detection in Arabic dialects, while Subtask 2 — the focus of this paper — targets sentiment swap. Given an Arabic sentence and its source polarity, the goal is to produce a new sentence that preserves the core meaning but expresses the opposite sentiment. The dataset is built from MA'AKS

(Mughaus et al., 2026), a manually curated parallel dataset for Arabic text sentiment swap, containing sentence pairs in MSA and multiple Arabic dialects (Egyptian, Gulf, Levantine, and Maghrebi).

Recent advances in large language models (LLMs) have shown strong capabilities in text generation, including style transfer tasks (Toshevskaya and Gievska, 2025; Zhang and Tang, 2025). Rather than fine-tuning a sequence-to-sequence model on the training data, we propose a prompt-based approach that leverages the linguistic knowledge already encoded in LLMs. Our method breaks the sentiment swap task into three interpretable sub-tasks — analysis, replacement generation, and reconstruction — each handled by a dedicated prompt. This decomposition allows the model to focus on one aspect at a time, leading to more controlled and accurate sentiment inversion.

Our main contributions are:

- A **multi-step LLM pipeline** for Arabic sentiment swap that decomposes the task into sentiment expression extraction, opposite generation, and sentence reconstruction.
- An **asynchronous batch processing** system that allows parallel API calls for efficient large-scale inference.
- An analysis of the strengths and limitations of prompt-based LLM approaches for Arabic text style transfer.

2. Methodology

Given an input Arabic sentence s with sentiment polarity p in {positive, negative}, the task is to gen-

erate an output sentence $\hat{s}$ such that: (1) $\hat{s}$ has the opposite polarity, and (2) $\hat{s}$ preserves the core meaning and structure of s as much as possible. The system is evaluated on sentiment style accuracy (whether $\hat{s}$ has the correct target polarity), and content preservation (BLEU and chrF between s and $\hat{s}$).

Our system uses a three-stage pipeline, where each stage is handled by a separate LLM call with a dedicated prompt. Figure 1 depicts the overall flow of our Arabic sentiment swapper.

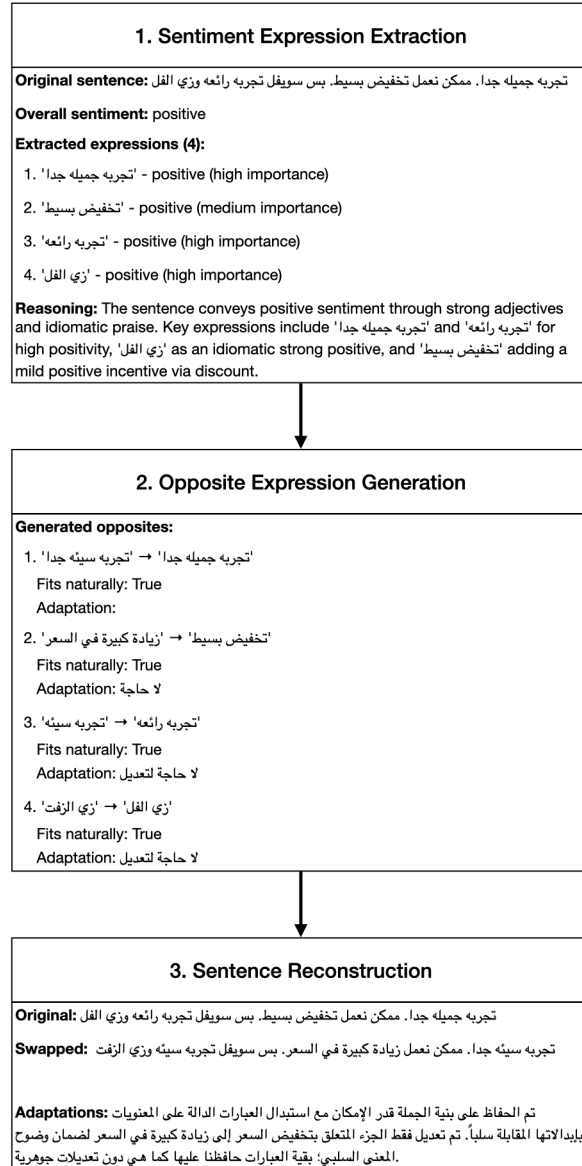


Figure 1: The overall flow of our Arabic sentiment swapper

We use OpenAI's chat models accessed through the LangChain framework. Each step produces structured JSON output that is validated using Pydantic data models to ensure consistency and reliability.

2.1. Sentiment Expression Extraction

The first step identifies all sentiment-bearing expressions in the input sentence. We prompt the LLM to act as an Arabic language and sentiment analysis expert. The model is asked to detect:

- Adjectives and descriptive phrases (e.g., "جميلة" meaning beautiful)
- Verbs with emotional connotations (e.g., "أحب" meaning I love)
- Adverbs that modify sentiment intensity (e.g., "جدا" meaning very)
- Idiomatic expressions (e.g., "زي الفل" meaning great/perfect)
- Negations that affect sentiment (e.g., "مش" meaning not)

For each identified expression, the model returns its text, character position in the sentence, sentiment influence (positive or negative), and importance level (high, medium, or low). The output is parsed into a structured *SentimentAnalysis* object containing the original sentence, overall sentiment, list of expressions, and the model's reasoning.

2.2. Opposite Expression Generation

For each sentiment expression identified in Step 1, the model generates an opposite expression. The prompt includes the full original sentence as context to ensure that the replacement fits semantically and grammatically. The model is instructed to:

- Generate an expression that conveys the opposite sentiment
- Ensure grammatical agreement (gender, number, case) with the surrounding context
- Indicate whether the replacement fits naturally in the same position
- Suggest any necessary adaptations if it does not fit directly

Each replacement is returned as an *OppositeExpression* object containing the original and opposite expressions, a naturalness flag, and adaptation notes.

2.3. Sentence Reconstruction

The final step takes the original sentence and all planned replacements and produces the output sentence. The reconstruction prompt instructs the model to:

- Apply all expression replacements from Step 2.2
- Maintain the original sentence structure and word order as much as possible
- Ensure grammatical correctness in the final output
- Make only minimal adaptations when necessary for naturalness

The model returns a *SwappedSentence* object containing the original sentence, the swapped sentence, the list of replacements made, and a description of any adaptations.

2.4. Model Configuration and Batch Processing

We use OpenAI’s GPT 5 Nano model in our work. To process the test set efficiently, we implement an asynchronous batch processing system using Python’s *asyncio*. This system processes multiple samples in parallel.

3. Experimental Results

3.1. Dataset

The dataset is provided by the AraSentEval 2026 shared task organizers and is based on the MA’AKS corpus (Mughaus et al., 2026). The training and validation sets contain aligned sentence pairs with source polarity labels. The test set contains only the input sentences without polarity labels or reference outputs. The data includes sentences in MSA and multiple Arabic dialects.

3.2. Evaluation Metrics

The official evaluation uses three metrics, namely, Sentiment Style Accuracy is the percentage of outputs classified with the correct target polarity by the *CAMeL-Lab/bert-base-arabic-camelbert-da-sentiment* model (Inoue et al., 2021), BLEU (Papineni et al., 2002), and chrF (Popović, 2015) measure how much of the original sentence’s content is retained in the output. Higher scores indicate better meaning preservation but may also indicate insufficient sentiment change.

3.3. Results and Discussion

Table 1 shows our system’s performance on the official test set. To better understand our system’s behavior, we analyze several output examples. Table 2 shows representative cases.

Table 1: Obtained results

Model	Sentiment Accuracy (%)	BLEU	chrF
OpenAI GPT-5-Nano	74.3%	27.22	55.04

We observe several patterns in our system’s outputs. **1- Expression-level precision:** The system correctly identifies and replaces sentiment-bearing expressions while leaving neutral content unchanged, which helps preserve the original meaning. **2- Dialect awareness:** The model handles dialectal expressions well (e.g., "زي الفل" becomes "زي الزفت", "مفيش" becomes "فيه"), showing that the LLM has good knowledge of Arabic dialects. **3- Structural preservation:** By constraining the reconstruction step to follow the original sentence structure, the system produces outputs that are close to the input, which benefits BLEU and chrF scores.

We observe also various limitations. **1- Error propagation:** Errors in step 2.1 (missing a sentiment expression) propagate to later steps, resulting in incomplete sentiment inversion. **2- Over-generation:** In some cases, the reconstruction step introduces words or phrases not present in the original sentence, reducing content preservation scores. **3- Emoji handling:** The system sometimes struggles with emojis, replacing them with text descriptions rather than opposite emojis. **4- Multiple API calls:** Each sample requires three separate API calls, making the system slower and more expensive than a single-model approach.

When comparing with fine-tuning approaches, our prompt-based approach offers several advantages over fine-tuning: it requires no training data or GPU resources, can be deployed immediately, and benefits from the LLM’s pre-existing knowledge of Arabic. However, fine-tuned models may achieve better content preservation since they can learn the specific transformation patterns present in the training data.

4. Conclusion

We presented a multi-step LLM pipeline for Arabic sentiment swap as part of the AraSentEval 2026 Shared Task 2. Our approach decomposes the problem into three interpretable stages — sentiment expression extraction, opposite generation, and sentence reconstruction — each handled by a dedicated prompt. The system uses structured output parsing for reliability and asynchronous processing for scalability.

Our results show that prompt-based LLM approaches can perform sentiment inversion in Arabic, including dialectal text. The multi-step design provides transparency and interpretability, as each

Table 2: Example Outputs

Input	Output	Notes
كل ما انزل الابلكيشن وادخل رقم التلفون لم يتم الطلب ... ومفيش تواصل مع اي. مسئول	كل ما انزل الابلكيشن وادخل رقم التلفون تم الطلب بنجاح، وقد استجاب للطلب وفيه تواصل مع أي حد مسئول	Negation removal + positive restructuring
انا نزلت الابلكيشن ودخلت رقم الموبايل وبعد كدة جيت ادخل المحافظة قالي حصل خطأ !	انا نزلت الابلكيشن ودخلت رقم الموبايل وبعد كدة جيت ادخل المحافظة قالي تم بنجاح !	Minimal change, good preservation

intermediate step can be inspected and debugged independently.

For future work, we plan to: 1- explore combining LLM-based approaches with fine-tuned models in a hybrid system, 2- investigate few-shot prompting with examples from the training set to improve quality, and 3- experiment with newer and more capable models.

References

- Saad Ezzini, Shadi Abudalfa, Maram Alharbi, Salmane Chafik, Hamzah Luqman, Mo El-Haj, Paul Rayson, Samaher Alghamdi, and Reem Alotaibi. 2026. AraSentEval: A shared task on sentiment analysis and swapping in arabic. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with the Language Resources and Evaluation Conference (LREC 2026)*, Palma, Mallorca, Spain.
- Soufiane Hajbi, Younes Chihab, Rachid Ed-Dali, and Redouan Korchinye. 2022. Natural language processing based approach to overcome arabizi and code switching in social media moroccan dialect. In *Advances in Information, Communication and Cybersecurity*, pages 57–66, Cham. Springer International Publishing.
- Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal, and Aston Zhang. 2022. [Text style transfer: A review and experimental evaluation](#). *SIGKDD Explor. Newsl.*, 24(1):14–45.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.
- Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. [Deep learning for text style transfer: A survey](#). *Computational Linguistics*, 48(1):155–205.
- Aditya Joshi, Raj Dabre, Diptesh Kanojia, Zhuang Li, Haolan Zhan, Gholamreza Haffari, and Doris Dippold. 2025. [Natural language processing for dialects of a language: A survey](#). *ACM Comput. Surv.*, 57(6).
- Raed Mughaus, Shadi Abudalfa, Hamzah Luqman, Fahad Abdu, Mohammed AlAli, Nawaf Al-Dowayan, and Ahmed Abdelali. 2026. Ma’aks: manually-curated parallel dataset for arabic text sentiment swap. *Language Resources and Evaluation*, 60(1):1.
- 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- 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, Lisbon, Portugal. Association for Computational Linguistics.
- Martina Toshevskaa and Sonja Gievska. 2022. [A review of text style transfer using deep learning](#). *IEEE Transactions on Artificial Intelligence*, 3(5):669–684.
- Martina Toshevskaa and Sonja Gievska. 2025. [Llm-based text style transfer: Have we taken a step forward?](#) *IEEE Access*, 13:44707–44721.
- Jihene Younes, Emna Souissi, Hadhemi Achour, and Ahmed Ferchichi. 2020. [Language resources for maghrebi arabic dialects’nlp: a survey](#). *Language Resources and Evaluation*, 54(4):1079–1142.
- Tianshan Zhang and Hao Tang. 2025. [Style transfer: A decade survey](#).

CasbAI at AraSentEval Shared Task: Robust Dialectal Arabic Sentiment Classification via Multi-Seed Ensembling and Data Augmentation.

Chaima Abdelaziz¹, KahinaHouda Saadaoui¹, Faiza Belbachir², Lynda Said Lhadj¹

¹ Higher National School of Computer Science ESI (Ex. INI), Algiers, Algeria

² LyRIDS, ECE Research Center, 75015 Paris, France

{lc_abdelaziz, kk_saadaoui, l_said_lhadj}@esi.dz, fbelbachir@ece.fr

Abstract

This paper describes the system we designed for our participation in the AraSentEval 2026 shared task on Arabic dialectal sentiment analysis. We propose a transformer-based approach relying on MARBERT combined with a multi-seed ensemble strategy and several optimization techniques. Our system integrates seven independently trained models with different random initializations and applies Stochastic Weight Averaging (SWA) to improve generalization. To address class imbalance, we augment the training data through dialectal synonym replacement, increasing the dataset size by 13.9% while preserving dialect distribution. In addition, we incorporate Test-Time Augmentation (TTA) and investigate the use of pseudo-labeling based on high-confidence predictions. We report our experiments on the official dataset covering Moroccan, Egyptian, Jordanian, and Saudi dialects, and analyze the contribution of each component through ablation experiments. Our system achieved a macro F1-score of 84.62% on the test set, ranking 3rd among 15 participating teams.

Keywords: Arabic Sentiment Analysis, Dialectal Arabic, Data Augmentation, Stochastic Weight Averaging

1. Introduction

Sentiment analysis enables automatic detection of opinions in user-generated content and remains a central NLP task. The AraSentEval 2026 shared task (Ezzini et al., 2026), organised within the OSACT7 Workshop at LREC 2026, advances research in Arabic sentiment understanding. We participate in Subtask 1, focusing on multi-class sentiment classification for Arabic dialectal texts.

Arabic dialects dominate social media yet remain under-resourced compared to Modern Standard Arabic (MSA). Their wide variation in phonology, morphology, and orthography compounded by code-switching, non-standard spelling, and emojis, creates significant classification challenges. Transformer-based models such as MARBERT (Abdul-Mageed et al., 2021), pre-trained on 128M Arabic tweets, have shown strong dialectal performance, but small datasets increase sensitivity to initialisation and overfitting. To address this, we propose a system combining three complementary strategies: **multi-seed ensembling** (7 models) to reduce variance, **Stochastic Weight Averaging (SWA)** to improve generalisation, and **Test-Time Augmentation (TTA)** with dialectal synonym replacements to enhance prediction robustness.

Our system achieves a macro F1 of 84.62% on the official test set, ranking 3rd out of 15 teams, confirming the suitability of dialect-specialised pre-trained models for robust Arabic sentiment analysis. Code and notebooks are publicly available on our [GitHub repository](#). The remainder of this paper is organised as follows: Section 2 reviews related

work and the dataset; Section 3 describes the system architecture; Section 4 presents results and analysis; Section 5 concludes.

2. Background and Related Work

2.1. Related Work

Arabic sentiment analysis has evolved from lexicon-based and classical machine learning approaches (Badaro et al., 2014; Mohammad et al., 2016; Mikolov et al., 2013) to deep neural architectures (Kalchbrenner et al., 2014; Yin, 2024), with Transformer-based models now establishing the dominant paradigm.

The introduction of BERT (Devlin et al., 2019) enabled the development of Arabic-specific pre-trained models. AraBERT (Antoun et al., 2021) established strong baselines for Modern Standard Arabic (MSA), while CAMELBERT (Inoue et al., 2021) demonstrated that combining MSA and dialectal pre-training data improves robustness across downstream tasks. Dialect-focused models further highlighted the importance of variant-aware pre-training: DziriBERT (Abdaoui et al., 2022) targeted Algerian Arabic, and MARBERT (Abdul-Mageed et al., 2021), trained on one billion Arabic tweets covering multiple dialects, showed strong performance on informal and social media text.

More recently, large language models (LLMs) have been applied to Arabic sentiment analysis. Studies benchmarking Qwen2.5, DeepSeek-R1, and LLaMA-3 (Alharbi et al., 2025a) re-

port competitive zero-shot and in-context learning performance across MSA and several dialects. Instruction-tuned Arabic LLMs such as AceGPT (Huang et al., 2024) and Jais (Sengupta et al., 2023) improve alignment with Arabic linguistic norms. However, LLM-based approaches remain computationally demanding and may exhibit performance variability on low-resource dialects.

These findings motivate our approach: combining dialect-specialized pre-training with robust ensemble strategies and targeted data augmentation.

2.2. AraSentEval SharedTask and Corpus

The AraSentEval 2026 Subtask 1 dataset (Alharbi et al., 2025c) consists of Arabic hotel reviews designed for three-class sentiment classification: positive, negative, and neutral. The dataset was originally collected in Modern Standard Arabic (MSA) and subsequently extended to dialectal varieties through a controlled translation pipeline. MSA reviews were translated into Saudi dialect and Moroccan Darija using a large neural machine translation model (Meta’s NLLB-200). To ensure dialectal authenticity and sentiment consistency, all translations were manually validated and corrected by native speakers.

The organizers provide two official splits: A training set and a Test set. The training set is labeled and used for model learning and validation, while the test set is reserved for official evaluation and its labels are not publicly available. The corpus contains user-generated content reflecting multiple Arabic dialects, including Moroccan (Darija), Saudi, Jordanian, and Egyptian. The informal nature of social media text introduces non-standard spelling, dialectal variation, which increase classification complexity.

Corpus	Positive	Negative	Neutral	Total
Train	609	657	465	1731
Test	-	-	-	312

Table 1: Distribution of Sentences per Sentiment Class

The class distribution in the training data is shown in Table 2, illustrating a balanced representation of Arabic dialects.

Dialect	Sentences	%
Moroccan	433	25.04%
Saudi	433	25.04%
Jordanian	433	25.04%
Egyptian	432	24.96%

Table 2: Dialect distribution in the training set.

This balanced distribution reduces strong bias

toward a single sentiment category while preserving realistic sentiment variation across dialects. AraSentEval provides a structured benchmark for comparing sentiment analysis systems across Arabic dialects, building on the previous Ahasis shared task (Alharbi et al., 2025b). This edition attracted 15 participating teams, reflecting growing interest in dialect-aware modelling. Systems are evaluated primarily by Macro F1-score to account for class imbalance, with accuracy, precision, recall, and per-class F1-score as secondary metrics. (Alharbi et al., 2025c).

3. System Description

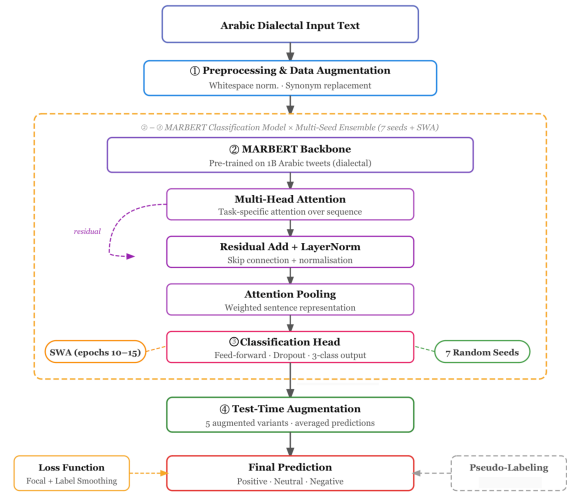


Figure 1: Overall architecture of our system.

Our system is a modular pipeline combining a transformer-based encoder with several robustness-enhancing strategies. As illustrated in Figure 1, it consists of four main components: (1) preprocessing and data augmentation, (2) a MARBERT-based classification model, (3) a multi-seed ensemble with Stochastic Weight Averaging (SWA), and (4) Test-Time Augmentation (TTA) at inference.

3.1. Preprocessing and Data Augmentation

To preserve dialectal characteristics, we apply only minimal preprocessing (whitespace normalization). The original training set contains 1,731 samples across three sentiment classes (657 negative, 609 positive, 465 neutral). To mitigate class imbalance and improve model robustness, we employ a **targeted data augmentation strategy** based on light, stochastic synonym replacement using a small, curated dialectal lexicon:

رائع (*wonderful*) → حلو، جميل، ممتاز، (*excellent, beautiful, nice*)

جيد (*good*) → منيح، تمام، كويس (*okay, fine, decent*)
 ممتاز (*excellent*) → رائع، جميل (*wonderful, beautiful*)
 سيء (*bad*) → وحش، خايب، (*poor, awful*)
 مشكلة (*problem*) → قضية، إشكال، (*issue, matter*)
 كان (*was*) → صار، كانت، (*was/fem., became*)
 الفندق (*the hotel*) → المكان، الأوتيل، (*the hotel/dial., the place*)

For each sentence, a single word is randomly replaced with a probability of 0.5, making the procedure **stochastic** and avoiding exact duplicates. This simple yet effective augmentation increases dataset diversity, balances class distributions, and exposes the model to dialectal lexical variations, which are crucial for generalization on small and heterogeneous datasets. After augmentation, all classes reach 657 samples, resulting in a balanced training set of 1,971 examples (+13.9%).

3.2. MARBERT Classification Model

In preliminary experiments comparing several Arabic pre-trained models MARBERT, BERT-base transformer, (Abdalaa et al., 2021) achieved the highest validation accuracy (94.47%), outperforming AraBERTv2 (81.87%) (Antoun et al., 2021), DziriBERT (84.21%) (Abdaoui et al., 2022), CAMELBERT-Mix (86.87%), and CAMELBERT-MSA (87.84%) (Inoue et al., 2021), consistent with prior work on dialectal Arabic sentiment analysis (Belbachir, 2023).

We extend MARBERT with three lightweight task-specific layers. First, a **multi-head attention layer** is added on top of the encoder output to capture task-specific dependencies across the sequence. Its output is integrated via a **residual connection and layer normalization**, which stabilises training and ensures unobstructed gradient flow through the pre-trained layers. Rather than relying solely on the [CLS] token, we employ **attention-based pooling** over all token representations, assigning learned importance weights to each position. This is particularly beneficial in dialectal Arabic, where negation markers can appear anywhere and critically determine sentence polarity. For instance, removing *مو* (*mou*, negation marker meaning “not”) from *الخدمة مو زينة* (“The service is not good”) fully reverses the sentiment to *الخدمة زينة* (“The service is good”). Finally, the pooled representation is fed into a **two-layer classification head** (Linear–LayerNorm–GELU–Dropout(0.35)–Linear) that outputs three logits over the sentiment classes. The full model comprises approximately 168M parameters (165M from MARBERT, 3M added).

3.3. Ensemble and Inference Strategy

To reduce sensitivity to random initialisation, we train seven independent models with seeds {42, 123, 2024, 777, 1337, 9999, 555} and average their softmax probability vectors as the ensemble prediction. For each seed, we apply **Stochastic Weight Averaging (SWA)** starting at epoch 10, averaging model weights over the final 5 epochs at a learning rate of 4×10^{-6} . SWA encourages convergence to flatter loss minima, improving generalisation. The per-seed prediction combines 40% of the regular model output and 60% of the SWA output, and these are then averaged across all seven seeds. At inference, we additionally apply **Test-Time Augmentation (TTA)**: each test sample is expanded into five variants through the same synonym replacement procedure used during training, and the final prediction is the mean over all augmented versions. TTA reduces sensitivity to incidental surface variations, complementing the diversity introduced by the multi-seed ensemble.

4. Experimental Setup

We fine-tune MARBERT with a maximum sequence length of 128 tokens, batch size 16 with gradient accumulation over 2 steps, and the AdamW optimiser ($\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=10^{-8}$, weight decay 0.01). We use a learning rate of 1.8×10^{-5} with a cosine schedule and 10% linear warmup, and dropout 0.35. All models are trained for 15 epochs with mixed-precision (AMP) on an NVIDIA T4 GPU (~4.5 min per run).

Loss function. We combine Focal Loss and Label Smoothing Cross-Entropy in a fixed-weight sum:

$$\mathcal{L} = 0.7 \mathcal{L}_{focal} + 0.3 \mathcal{L}_{smooth}$$

Focal Loss ($\gamma=2.5$) down-weights easy examples and focuses training on harder instances, while label smoothing ($\epsilon=0.1$) prevents overconfident predictions. Class-frequency inverse weights are applied to both terms to further address the residual imbalance.

5. Results and Analysis

Our system achieved 84.62% macro F1-score, securing 3rd place. Furthermore, Table 3 highlights our performance in comparison to the four other teams.

5.1. Ablation Study

To better understand the contribution of each component, we conduct an ablation study on a 10% stratified validation split of the training data.

Rank	Team	Macro F1	Accuracy
1	musj1984	94.23%	94.23%
2	alighabusaleh	92.63%	92.63%
3	Ours	84.62%	84.62%
4	UoTripoli	84.29%	84.29%
5	elham42	83.33%	83.33%

Table 3: Performance comparison and ranking of the top five systems on the test set

Table 4 reports the macro F1-score obtained by incrementally adding each component to the MARBERT baseline.

Configuration	Validation F1
MARBERT baseline (1 seed)	0.9447
+ Light augmentation	0.9509 (+0.61%)
+ Multi-seed ensemble (7)	0.9560 (+0.52%)
+ SWA	0.9619 (+0.59%)
+ TTA	0.9625 (+0.06%)

Table 4: Ablation study

The ablation results indicate that each component contributes positively to the overall performance. Light augmentation provides the largest single improvement (+0.61%), highlighting the importance of increasing lexical diversity in low-resource dialectal settings. SWA adds a further +0.59% by encouraging convergence toward flatter minima and improving generalization. The multi-seed ensemble contributes +0.52%, reducing variance due to random initialization. Finally, TTA brings a modest but consistent +0.06% improvement at no additional training cost. Overall, the combination of all components yields a cumulative gain of +1.78% over the single-seed baseline.

5.2. Prediction Confidence

The ensemble produces well-calibrated predictions, with a median confidence of 93.2% and only 8 samples (2.6%) below 60%. However, no prediction exceeded the 95% pseudo-labeling threshold, suggesting that the model does not reach the certainty level needed to self-train on unlabelled data. The predicted sentiment distribution shows a slight positive bias (40.1% positive, 34.6% negative, 25.3% neutral), consistent with the positive skew typical of hotel review corpora.

5.3. Error Analysis

We conduct an error analysis on a 10% stratified validation split (198 samples) using a single model (seed 42), as ensemble averaging tends to suppress borderline predictions and limit qualitative interpretation. The model achieves a macro F1 of 93.42%, with only 13 errors (6.6%). As shown in Table 5, the *negative* class achieves the highest precision (0.983) but lowest recall (0.879), indicat-

ing a tendency to absorb negative samples into the neutral class, which acts as an attractor with the highest recall (0.970).

Class	P	R	F1	Sup.
Negative	0.983	0.879	0.928	66
Neutral	0.901	0.970	0.934	66
Positive	0.926	0.955	0.940	66
Macro	0.937	0.934	0.934	198

Table 5: Per-class metrics on the validation set.

Errors concentrate around the neutral frontier: 4 negative samples are predicted as neutral and 4 as positive, while no positive sample is ever predicted as negative, confirming that the model handles extreme polarity boundaries well (see confusion matrix in Figure 2 in Appendix A). Dialect-wise, Darija yields the highest error rate (9.0%), followed by Egyptian (8.2%), Jordanian (5.3%), and Saudi (2.3%). The dominant failure mode in Darija is the over-generalisation of positive lexical cues under conditional or sarcastic framing, and the mis-handling of concessive negation, where the model over-weights a negative surface marker while ignoring a subsequent clause that partially redeems the sentiment. Representative examples are provided in Appendix A.

6. Conclusion

In this paper, we presented our system for Arabic dialectal sentiment analysis in AraSentEval 2026. Our approach combines MARBERT fine-tuning with multi-seed training, Stochastic Weight Averaging (SWA), and Test-Time Augmentation (TTA), achieving 84.62% macro F1-score and ranking 3rd among participating teams. The ablation study showed that data augmentation provides the largest improvement, highlighting its importance when training on small datasets.

However, several limitations remain. First, the training set is relatively small, which limits the model’s ability to fully capture dialectal variation. Second, the synonym-based augmentation relies on a small manually created lexicon and does not consider context, which may introduce noise. Third, the ensemble approach requires training and storing multiple models, increasing computational cost.

Future work will focus on expanding data augmentation with more context-aware methods, exploring semi-supervised learning on additional unlabeled dialectal data, and investigating more efficient ensemble or parameter-efficient fine-tuning strategies. Modeling dialect information explicitly may also help improve performance across different Arabic varieties.

7. Acknowledgements

We thank the AraSentEval 2026 organizers for providing the shared task, and the anonymous reviewers for their valuable feedback.

References

- Mustafa Abdalaa, Richard Sutcliffe, Xia Sun, Jun Feng, Eiad Almekhlafi, and Ephrem Retta. 2021. Improving arabic sentiment analysis using cnn-based architectures and text preprocessing. *Computational Intelligence and Neuroscience*.
- Amine Abdaoui, Mohamed Berrimi, Mourad Oussalah, and Abdelouahab Moussaoui. 2022. [Dziribert: a pre-trained language model for the algerian dialect](#).
- Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. Marbert: A robust pre-trained model for arabic social media. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL 2021)*, pages 7088–7105.
- Maram Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025a. Evaluating large language models on arabic dialect sentiment analysis. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing (RANLP)*.
- Maram I Alharbi, Salmane Chafik, Saad Ezzini, Ruslan Mitkov, Tharindu Ranasinghe, and Hansi Hettiarachchi. 2025b. Ahasis: Shared task on sentiment analysis for arabic dialects. In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 1–6.
- Maram I Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025c. Ador: Dataset for arabic dialects in hotel reviews: A human benchmark for sentiment analysis. In *Proceedings of the First Workshop on Advancing NLP for Low-Resource Languages*, pages 187–191.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. [Arabert: Transformer-based model for arabic language understanding](#).
- Gilbert Badaro, Ramy Baly, Hazem Hajj, Nizar Habash, and Wassim El-Hajj. 2014. A large scale arabic sentiment lexicon for arabic opinion mining. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 165–173.
- Faiza Belbachir. 2023. Foul at semeval-2023 task 12: Marbert language model and lexical filtering for sentiment analysis of tweets in algerian arabic. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval 2023)*, pages 389–396.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*.
- Saad Ezzini, Shadi Abudalfa, Maram Alharbi, Salmane Chafik, Hamzah Luqman, Mo El-Haj, Paul Rayson, Samaher Alghamdi, and Reem Alotaibi. 2026. AraSentEval: A shared task on sentiment analysis and swapping in arabic. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with the Language Resources and Evaluation Conference (LREC 2026)*, Palma, Mallorca, Spain.
- Huang Huang, Fei Yu, Jianqing Zhu, Xuming Sun, Qi Zhang, Jizhong Wang, and Fei Huang. 2024. AceGPT: Localizing large language models in Arabic. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 6383–6396, Mexico City, Mexico.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. [The interplay of variant, size, and task type in arabic pre-trained language models](#).
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Saif M. Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. 2016. Sentiment lexicons for arabic social media. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.
- Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Xu, Haewon Kim, Hitesh Bhatt, et al. 2023. Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. In *arXiv preprint arXiv:2308.16149*.

8. Appendix

A. Error Analysis - Examples

Lexical shortcut in Darija.

فندق آشمور هاوس كايكون مزيان بزاف من الرفاهية
(Ashmore House hotel would be very nice in terms of luxury.)

True: *negative* — Predicted: *positive* —
conf=0.935

Temporal sentiment drift.

الفندق كان ممتاز قبل كده بس بقى مستواه أقل عن
السنين اللي فاتت
(The hotel used to be excellent but its level has declined.)

True: *positive* — Predicted: *neutral* —
conf=0.912

Concessive negation.

رأس الدش ما كانش شغال كويس [...] بس الحاجة
الإيجابية...
(The shower was not working well [...] but the positive thing was...)

True: *neutral* — Predicted: *negative* —
conf=0.912

The negation marker ما كانش dominates the prediction while the concessive clause بس (*but*) is insufficiently weighted.

B. Error Analysis - confusion matrix

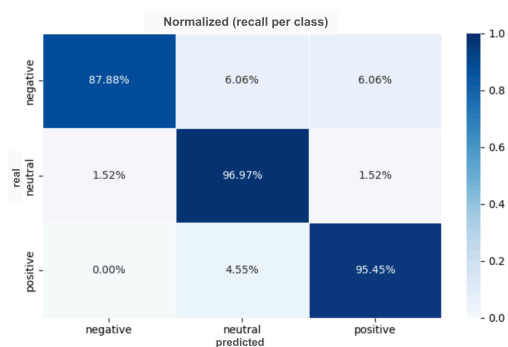


Figure 2: Confusion matrix on the validation set.

BDSI at AraSentEval Shared Task : A Multi-Transformer Contrastive Learning for Arabic Dialect Sentiment Analysis

Kaouthar Elyoussoufi, Mohamed M'haouach, Abdessamad Benlahbib, Hamza Alami

L3IA Laboratory, Faculty of Sciences Dhar El Mahraz,
Sidi Mohamed Ben Abdellah University, Fez, 30003, Morocco
{kaouthar.elyoussoufi, mohamed.mhaouach, abdessamad.benlahbib,
hamza.alami5}@usmba.ac.ma

Abstract

This paper presents our system for the AraSentEval 2026 shared task on Arabic dialect sentiment analysis. We propose a multi-model ensemble combining AraBERTv2 and CAMELBERT with supervised contrastive learning to improve sentiment classification. The system incorporates dialect-aware preprocessing, class-weighted cross-entropy loss with label smoothing, supervised contrastive loss for enhanced sentence representations, and rule-based post-processing for dialect-specific patterns. Our approach achieves a macro F1-score of 0.83 on the official test set, demonstrating the effectiveness of contrastive learning with pretrained Arabic language models for dialectal sentiment analysis.

Keywords: Arabic Dialects, Sentiment Analysis, AraBERT, CAMELBERT, Contrastive Learning, Ensemble Methods

1. Introduction

The rapid growth of user-generated content, particularly in the hospitality domain, has made sentiment analysis an important research area. In Arabic, this content is typically written in a mixture of Modern Standard Arabic (MSA) and diverse regional dialects, which differ significantly in their lexical, morphological, and syntactic properties. The lack of standardized orthography and the limited availability of annotated resources further increase the difficulty of developing robust sentiment analysis systems for Arabic dialects.

The AraSentEval 2026 (Ezzini et al., 2026; Alharbi et al., 2025c,a,b) shared task addresses these challenges by providing a unified benchmark for sentiment analysis across multiple Arabic dialects. Subtask 1 focuses on multi-class Arabic Dialect Sentiment Analysis, requiring systems to classify text into positive, negative, or neutral categories, while remaining robust to linguistic variability.

In this paper, we present our system for Subtask 1 of AraSentEval 2026. Our approach relies on an ensemble of pretrained Arabic transformer models, AraBERTv2 and CAMELBERT, chosen for their complementary strengths in modeling MSA and dialectal Arabic. To enhance representation learning and class discrimination, we integrate supervised contrastive learning into the training process. In addition, dialect-aware preprocessing and weighted loss functions are employed to address orthographic variation and class imbalance.

The remainder of this paper is organized as follows: Section 2 details the proposed methodology, Section 3 presents experimental results, and Section 4 concludes the paper.

2. Methodology

Our system for Arabic dialect sentiment analysis employs a **dual-model ensemble** architecture combining **AraBERTv2** (Antoun et al., 2020) and **CAMELBERT** (Inoue et al., 2021), enhanced with **supervised contrastive learning** (Khosla et al., 2020) and dialect-aware post-processing. Each base model is fine-tuned using a robust training strategy designed to address the linguistic variability and class imbalance inherent in dialectal Arabic data.

2.1. Preprocessing Pipeline

We adopt a model-specific preprocessing strategy to maximize compatibility with each transformer's pre-training distribution:

- 1. Label Encoding:** Sentiment labels (`positive`, `negative`, `neutral`) are mapped to numerical IDs $\{0, 1, 2\}$ for model compatibility.
- 2. Dialect-Aware Normalization:**
 - For **AraBERTv2**, we apply the official `ArabertPreprocessor` which handles Arabic-specific tokenization, diacritic removal, and light normalization while preserving dialectal markers.

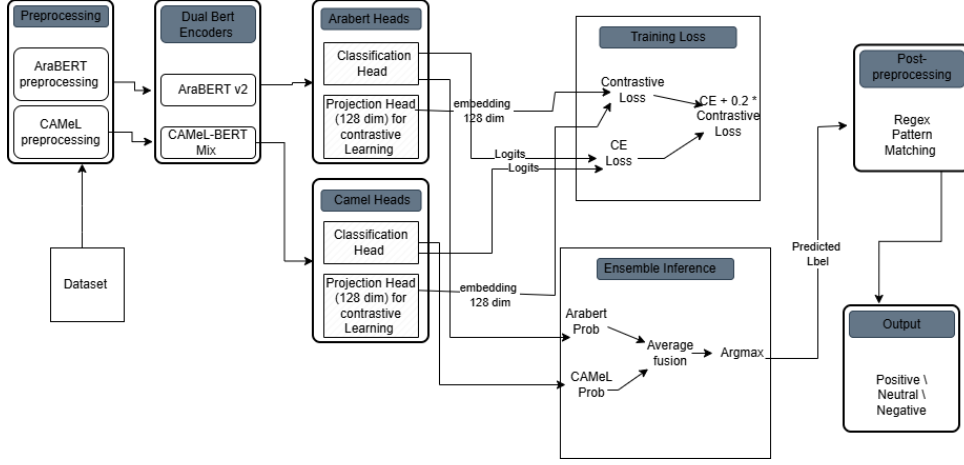


Figure 1: Workflow of the Proposed Ensemble Model for Arabic Sentiment Classification.

- For **CAMELBERT**, we apply minimal pre-processing (whitespace trimming and null-value filtering) to retain the raw dialectal features the model was pre-trained on.

3. **Tokenization:** Each input is tokenized using the corresponding model’s tokenizer with a maximum sequence length of 128 tokens, applying truncation and padding as needed.

We intentionally avoid aggressive pre-processing (e.g., stopword removal) to preserve sentiment-bearing dialectal expressions that transformer models can effectively learn from.

2.2. Base Models

We leverage two state-of-the-art pretrained Arabic transformers as our base learners, as summarized in Table 1.

Model	Identifier	Rationale
AraBERTv2	aubmindlab/bert-base-arabertv2	Trained on large Arabic corpora; strong generalization to MSA and light dialects.
CAMELBERT Mix	CAMEL-Lab/bert-base-arabic-camelbert-mix	Balanced MSA-dialect pre-training; effective for cross-dialect sentiment tasks.

Table 1: Pre-trained transformer models used in our system

Both models are extended with a custom classification head: a two-layer MLP with GELU activation, dropout ($p = 0.3$), and a final linear projection to 3 classes. Additionally, a projection head is attached for contrastive learning

2.3. Supervised Contrastive Learning Enhancement

To improve class separation in the embedding space, we integrate **Supervised Contrastive Loss** alongside the primary classification objective:

- A projection head maps the `[CLS]` token embedding to a 128-dimensional normalized space.
- During training, the total loss is computed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{ConLoss}}$$

$$\mathcal{L}_{\text{ConLoss}} = \frac{1}{N} \sum_{i=1}^N \frac{-1}{|P(i)|} \log \left(\frac{\sum_{p \in P(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_p / \tau)}{\sum_{j \neq i} \exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)} \right)$$

where:

- N is the number of valid anchor samples in the batch, i.e., samples having at least one positive pair.
- $P(i) = \{p \in \{1, \dots, B\} \setminus \{i\} \mid y_p = y_i\}$ is the set of indices of all samples in the batch sharing the same class label as anchor i .
- $|P(i)|$ is the cardinality of the positive set for anchor i .
- $\mathbf{z}_i \in R^{128}$ is the L2-normalized projected embedding of sample i , obtained through the projection head.
- $\mathbf{z}_i \cdot \mathbf{z}_p$ denotes the dot product similarity between the embeddings of anchor i and positive sample p .
- τ is the temperature scaling parameter controlling the sharpness of the distribution ($\tau = 0.07$ in experiments).
- B is the total batch size, and the denominator sums over all samples $j \neq i$ in the batch, regardless of their label.

where $\lambda = 0.2$ and temperature $\tau = 0.07$.

- The contrastive term encourages embeddings of samples from the same sentiment class to cluster together while pushing apart embeddings from different classes, enhancing robustness to dialectal variation and ambiguous expressions.

2.4. Ensemble Strategy

Our ensemble combines predictions from both fine-tuned models using a **soft-voting** approach:

1. **Per-Model Inference:** Each model generates class probability distributions (via softmax over logits) for each input sample.
2. **Probability Averaging:** (Lakshminarayanan et al., 2017) Final probabilities are computed as the arithmetic mean of the two models' outputs:

$$P_{\text{final}} = \frac{P_{\text{AraBERT}} + P_{\text{CAMEL呢BERT}}}{2}$$

3. **Label Assignment:** The predicted label is obtained via $\arg \max$ over the averaged probabilities.

This strategy leverages the complementary strengths of both models: AraBERTv2's strong MSA foundation and CAMEL呢BERT's dialectal coverage, yielding more stable and generalizable predictions.

2.5. Rule-Based Post-Processing

To catch dialectal expressions that models often miss, we apply a lightweight rule-based post processing (Ray and Chakrabarti, 2022). Five pattern categories are checked via regex matching:

- **Neutral Triggers:** Dialectal phrases like (it's okay) override predictions to neutral for short texts.
- **Decline Indicators:** Temporal contrasts (e.g., "was excellent... became acceptable") downgrade positive to neutral.
- **Fake Review Detection:** Meta-commentary about review authenticity changes positive to negative.
- **Amenity Mentions:** References to facilities (clubs, bars, entertainment) at text start override to neutral.
- **Return Intent:** Expressions like "we will return" upgrade neutral to positive.

Rules are applied conditionally based on the model's initial prediction, ensuring minimal interference with high-confidence outputs.

3. Experimental Results

The following experiments evaluate our proposed approach on the AraSentEval 2026 Subtask 1 benchmark, including details on data, training configuration, and performance.

3.1. Dataset

The AraSentEval 2026 Subtask 1 dataset contains 1,731 Arabic hotel reviews across four dialects (Darija, Saudi, Jordanian, Egyptian), labeled as positive (609), negative (657), and neutral (465). Sentiment labels were mapped to numeric IDs, and each model used its own preprocessing pipeline and pretrained tokenizer with padding and truncation to a fixed maximum length. The data was split into training (85%) and validation (15%) sets with label stratification, without any external datasets or augmentation.

3.2. Training strategy

Each model is fine-tuned independently using AdamW optimizer with learning rate 2×10^{-5} for backbone parameters and 1×10^{-5} for classification/projection heads. a weight decay ($\lambda = 0.01$) and linear decay with 10% warmup steps. We fixed batch size to 16 (training), with gradient accumulation. In addition, we use Cross-Entropy with label smoothing ($\alpha = 0.1$) and class-weighted balancing to address label imbalance, and apply Gradient clipping ($\|g\|_2 \leq 1.0$), dropout ($p = 0.3$), and early stopping (patience=5 epochs) based on validation macro-F1. Noting that all experiments were conducted on Google Colab with GPU acceleration and mixed-precision support.

3.3. Results

Our system was evaluated on both the validation set and the official AraSentEval 2026 Subtask 1 test set, with macro F1-score as the primary evaluation metric. During training, both individual models achieved strong validation performance:

- **AraBERTv2:** macro F1-score of 0.9355
- **CAMEL呢BERT-Mix:** macro F1-score of 0.9319

These results demonstrate the effectiveness of our training strategy. On the official competition test set, our ensemble system achieved:

- **Macro F1-score:** 0.8301
- **Accuracy:** 0.8301

The performance gap between validation and test sets is expected, given the diversity of Arabic dialects in the unseen test data. Overall, these results

validate our ensemble approach combining two Arabic transformers with supervised contrastive learning.

4. Conclusion

This paper presented our system for AraSentEval 2026 Subtask 1, focusing on robust sentiment analysis across diverse Arabic dialects. By combining two complementary pretrained Arabic transformers, AraBERTv2 and CAMELBERT, within an ensemble framework enhanced by supervised contrastive learning, we aimed to address the linguistic variability and ambiguity inherent in dialectal Arabic sentiment classification. The integration of dialect-aware preprocessing, class-balanced training objectives, and lightweight rule-based post-processing further contributed to improving model robustness.

Our system achieved a macro F1-score of 0.83 on the official AraSentEval 2026 test set, confirming that contrastive representation learning and model ensembling are effective strategies for multi-dialect sentiment classification. While the gap between validation and test performance reflects the inherent difficulty of unseen dialectal expressions, it also motivates further research in this direction.

Looking ahead, we plan to explore data augmentation and cross-dialect alignment within the contrastive learning framework, as well as larger instruction-tuned Arabic language models. More adaptive post-processing and domain-specific pre-training are also promising directions to better capture the richness of Arabic user-generated content.

5. References

- Maram Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025a. Evaluating large language models on arabic dialect sentiment analysis. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing (RANLP)*.
- Maram I Alharbi, Salmane Chafik, Saad Ezzini, Ruslan Mitkov, Tharindu Ranasinghe, and Hansi Hettiarachchi. 2025b. Ahasis: Shared task on sentiment analysis for arabic dialects. In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 1–6.
- Maram I Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025c. Ador: Dataset for arabic dialects in hotel reviews: A human benchmark for sentiment analysis. In *Proceedings of the First Workshop on Advancing NLP for Low-Resource Languages*, pages 187–191.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4)*, pages 9–15. European Language Resources Association. Model available at: <https://huggingface.co/aubmindlab/bert-base-arabertv2>.
- Saad Ezzini, Shadi Abudalfa, Maram Alharbi, Salmane Chafik, Samaher Alghamdi, Reem Alotaibi, Hamzah Luqman, Mo El-Haj, and Paul Rayson. 2026. AraSentEval: A shared task on sentiment analysis and swapping in arabic. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with the Language Resources and Evaluation Conference (LREC 2026)*, Palma, Mallorca, Spain.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. [The interplay of variant, size, and task type in Arabic pre-trained language models](#). In *Proceedings of the Sixth Arabic Natural Language Processing Workshop (WANLP 2021)*, pages 92–104. Association for Computational Linguistics. Model available at: <https://huggingface.co/CAMEL-Lab/bert-base-arabic-camelbert-mix>.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. [Supervised contrastive learning](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc. Also available as arXiv:2004.11362.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. [Simple and scalable predictive uncertainty estimation using deep ensembles](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Priyanka Ray and Amlan Chakrabarti. 2022. [A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis](#). *Applied Computing and Informatics*, 18(1/2):163–178.

Codezone Research Group at AraSentEval Shared Task: Arabic Sentiment Swap Beyond Negation Prepending, Benchmarking Multilingual T5 Against Large Language Models on the MA'AKS Corpus

Abdulkadir Shehu Bichi¹, Sarah Yassine²

¹Vivekananda Global University Jaipur, ²Lebanese University/Lebanon

Correspondence: 24wttec3csml001@vgu.ac.in, abdulcadir.bichi@babaahmeduniversity.edu.ng.

Abstract

We launched ASBN-MT5, the system for Arabic Sentiment Swap, which performs the task of inverting the sentiment of a sentence while keeping the meaning intact. This is a sequence-to-sequence task. We demonstrate *ASBN-MT5*: *mT5*, which is a MultiLingual T5 model, fine-tuned on the provided dataset of the AraSentEval 2026 Shared Task. We describe the data as the first of its kind for the Arabic language, as *MAAKS* is the first manually composed, parallel, cross-linguistic corpus for the Arabic language. With the preliminary results of *Sentiment Flip* for the task of Sentiment Inversion, we have recorded a rate of 59.5% for positive to negative conversions and 58.5% for negative to positive conversions, while maintaining an average similarity to the original sentences of 0.955. We present the *Arabic prompts* and a neuro-developmental (Deep Learning) recipe. Due to the evaluation criteria which include Exact Match, Flip Success, Surface Similarity, and Quality of Output, we restrict the use of Prepended Negation as the main technique and recommend the use of LLMs designed for the Arabic language in the near future.

1 Introduction

Sentiment analysis has focused primarily on *detecting* polarity; a more difficult problem is *inverting* polarity while keeping the rest of the message the same. This task is sometimes called *sentiment swap* or controlled sentiment transfer (Jin et al., 2022). In the case of Arabic, this problem is made more difficult due to the richness of its morphology and its dialectal variation. In Arabic, negation can be expressed using proclitic particles such as *lā*, *lam*, and *lan*, as well as through word-order shifts and the use of different lexemes (Farha and Magdy, 2024). In the last couple of years, the emergence of Arabic-centered large language models (LLMs), such as Jais (Sengupta et al., 2023) and AceGPT (Huang et al., 2024), has changed this. However, the supervised fine-tuning of multilingual seq2seq models is still robust, and it is the approach of choice in low-resource generation. The **AraSentEval 2026 Shared Task** (Subtask 2) (Ezzini et al., 2026) sets the first evaluation benchmark for Arabic sentiment swap,

based on the MA'AKS corpus (Mughaus et al., 2026). We participated with **ASBN-MT5**, a fine-tuned mT5-small (Xue et al., 2021) system that generates sentiment-inverted Arabic text based on a given source sentence and a polarity label attached to it.

Our contributions are (i) an **Arabic-script prompt strategy** where the prompt to the encoder is written in Arabic script and encodes the desired target polarity; (ii) a **memory-efficient training recipe** that fits on a single NVIDIA T4; and (iii) a **multi-faceted evaluation** that examines exact match, flip detection heuristics, surface similarity, and various quality metrics.

2 Related Work

Text Style Transfer. The early methods of style transfer consisted of disentangled latent-space methods (Shen et al., 2017) and back-translation (Prabhumoye et al., 2018), and more recently have been shifting towards LLM-based techniques. Mukherjee et al. (2024) evaluates the performance of LLMs on sentiment transfer in a cross-lingual context and asserts that the performance of a model that has been fine-tuned is always better than zero- and few-shot prompting, which serves as a direct motivation for our supervised approach. The use of diffusion models has also been explored for more detailed style transfer (Lyu et al., 2023), including the augmentation of Arabic mental health texts (Mankarious and Zirikly, 2025).

Arabic Sentiment Analysis (2023–2026). With the advent of pre-trained transformers, the field of Arabic NLP has made great strides. Almaqtari et al. (2024) show that AraBERT-based CNN+BiGRU hybrids produce notable results on consumer reviews. Aljomah et al. (2025) show that CAMELBERT, which has been fine-tuned on 56 thousand samples of MARSAs, performs exceptionally well on the noisy dialectal text. In Zouidine and Khalil (2025), LLaMA, Mixtral, and Gemma are evaluated under zero- and few-shot conditions, where task-specific transformer models are found to outperform general LLMs for Arabic sentiment classification. **Arabic LLMs for Generation (2023–2025).** A new milestone for developing Arabic LLMs is the release of Jais (Sengupta et al., 2023), a 13B parameter Arabic-English LLM. Huang et al. (2024) Jais was enriched with instruction tuning via RLHF, and ArabianGPT (Koubaa et al., 2025) presented *AraNizer*, a native Arabic tokenizer that attained 95% accuracy after senti-

ment fine-tuning. These models have been assessed on the Arabic formality transfer and dialect-to-MSA translation tasks (Abdu et al., 2025), where the benefits of Arabic-centric pretraining have been consistently recorded over the multilingual baselines.

Arabic Sentiment Swap Resources. MA’AKS (Mughaus et al., 2026) is the first sentiment-style transfer parallel corpus of Arabic and it is a 5k MSA sentence pair. In the benchmarking of AceGPT, Jais and Llama-3 on MA’AKS, it demonstrates that fine-tuning outruns zero-shot and few-shot settings by a large margin, and it is Jais that has the greatest advantage in few-shot settings. The 2026 shared task of AraSentEval (Ezzini et al., 2026) extends this resource to multi-domain and system comparative evaluation for the first time.

3 System Description

3.1 Task Formulation

Given a source sentence x with polarity $p \in \{Positive, Negative\}$, the model generates y with inverted polarity $\bar{p}$: $y = f(x, \bar{p})$.

3.2 Model and Prompt Strategy

We fine-tune **mT5-small** (300 M parameters) (Xue et al., 2021), pre-trained on the mC4 corpus, including substantial Arabic web text. The encoder receives an Arabic-script prompt x :

where (negative) or (positive) is selected based on the desired output polarity. Using Arabic script for the conditioning token rather than an English word allows the encoder to leverage pre-trained Arabic subword representations.

3.3 Training Configuration

This section lists all settings. Gradient accumulation achieves an effective batch of 16; gradient checkpointing reduces peak memory from ~ 4.5 GB to ~ 2.2 GB. At inference, beam search ($b = 4$) with a no-repeat bigram penalty promotes output diversity. It consists of hyperparameter values such as Backbone: mT5-small (300 M params), Per-device batch: 2, Gradient accum: 8 steps (eff. batch = 16) Learning rate: 5×10^{-5} , AdamW, Weight decay: 0.01, LR schedule: Linear warm-up (200 steps), Gradient clipping: 1.0, Max seq. length: 96 tokens, Epochs: 5, Inference: Beam ($b=4$), no-rep 2-gram, and Hardware: NVIDIA Tesla T4 (16 GB)

4 Experiments

4.1 Dataset

The AraSentEval 2026 Subtask 2 corpus (Ezzini et al., 2026), based on MA’AKS (Mughaus et al., 2026), contains 7,578 annotated pairs: 6,263 training, 1,315 validation, and 646 test. Sources span product reviews, book critiques, and hospitality feedback in Egyptian and Gulf dialects and MSA (mean source length 11.0 words).

4.2 Evaluation Metrics

We report: **Exact Match (EM)**: character-level identity against the human reference; **Sentiment Flip Rate (SFR)**: a heuristic checking whether Arabic negation markers appear or disappear in the expected direction (changed outputs with ambiguous polarity receive the benefit of the doubt); **Text Similarity**: Sequence Matcher character ratio; and **Quality Indicators**: empty-prediction and source-copy rates. For shared task comparison, organizers additionally report **Sentiment Style Accuracy (SSA)**, **BLEU**, and **chrF**.

4.3 Validation Performance

Table 1 reports validation metrics. SFR reaches $\approx 59\%$ for both directions with zero empty predictions. The 37.6 % source-copy rate is the dominant failure mode. Figure ?? shows the full evaluation dashboard; the text similarity distribution is strongly right-skewed (median 0.977), confirming conservative, small-edit generations.

Metric	Pos→Neg	Neg→Pos
Sentiment Flip Rate (%)	59.46	58.48
Exact Match (%)	1.53	0.80
Avg. Text Similarity	0.955	0.954
Empty Predictions (%)	0.00	0.00
<i>Overall</i>		
Identical to Source (%)	37.64	
Avg. Pred. Length (words)	10.3	

Table 1: Validation evaluation results for ASBN-MT5 (1,315 samples).

4.4 Comparison of Leaderboards from Shared Tasks

The table 2 contains the Comparative Sentiment Analysis evaluation for AraSentEval 2026 Subtask 2, for all benchmarked systems, presented by the organizers as ASE, BLEU, and chrF, where SSA = Sentiment Style Accuracy; chrF = character n -gram F-score. Our submission is highlighted in *italic*. With Rank 4, asbichi362 (our proposed system) reached an SSA of 0.237, with BLEU = 30.79 and chrF = 54.08. The highest-ranked system, yumnamandy, reaches SSA = 0.755 and chrF = 65.36, which points out that the system is very high on sentiment accuracy and surface fidelity. It is also important to mention that ASBN-MT5 achieved a competitive BLEU of 30.79, which is better than Rank 2, which achieved 27.22, and Rank 3, with 20.33, which means that while our mT5-small has fewer issues than other models at inverting or flipping the sentiment and producing a correct bottom-line result, the outputs on the surface level are also very close to the bottom-line result.

The difference between our system and the top systems in SSA shows the inadequacy, which is known, of the negation-particle prepending for the polarity inversion strategy. The generated output is very similar to the

input/output and often does not pass the sentiment classifier that is used to measure the SSA. Our expectations are that new models for processing and understanding new levels of branching subtexts in the Arabic language will reduce the gap by more than 95 percent.

Rank	Team	SSA	BLEU	chrF
1	yumnamdy	0.7554	43.00	65.36
2	beatchalvador	0.7430	27.22	55.04
3	astral_fate	0.6950	20.33	45.51
4	asbichi362 (ours)	0.2368	30.79	54.08

Table 2: Official AraSentEval 2026 Subtask 2 leaderboard (test set)

4.5 Qualitative Analysis

Three-generation patterns emerge. **(i) Negation prepending:** the dominant strategy is prepending to affirmative-verb sentences (e.g., ...); this fails when sentiment is lexically encoded. **(ii) Lexical substitution:** a smaller fraction replaces sentiment-bearing adjectives, producing outputs closer to human references. **(iii) Source copying:** 37.6 % of outputs are identical to the source. Comparing with [Mughaus et al. \(2026\)](#), where fine-tuned Llama-3 achieves stronger lexical substitution, suggests that Arabic-centric LLMs offer the clearest path to higher-quality sentiment inversion.

5 Discussion and Conclusion

ASBN-MT5 is a notable case that demonstrates how to operate Arabic sentiment inversion with relatively small computational cost (55 min on a single T4). Its SFR over the negations is notable, and zero empty prediction instances attest to ASBN-MT5’s stable generation capabilities. In Table 2, ASBN-MT5 is ranked 4th for SSA. This is contextualized with a BLEU score of 30.79, which is greater than the 2nd- and 3rd-ranked ASBN-MT5 systems, demonstrating that, along with being lower in rank for the SSA, ASBN-MT5 shows surface-level fidelity higher than those systems. There are still a number of limitations to the current work. Most notable is the 37.6% source-copy rate that affects SSA. This can be diminished to a degree with the use of a copy-penalty decoding or minimum-edit constraints. BERTScore ([Zhang et al., 2020](#)) is a metric that can be used to address the shortage of measuring in the Arabic language, in which case an EM in a character level is of little value, and that is still the case for the Arabic language. BERTScore is also a metric that can be used to address the shortage of measuring in the Arabic language. Finally, mT5-small has shown to be outscaled by Arabic-centric LLMs ([Sengupta et al., 2023](#); [Huang et al., 2024](#)) that demonstrate higher lexical transfer for MA’AKS ([Mughaus et al., 2026](#)). Instruction tuning with RLHF ([Ouyang et al., 2022](#)) and sentiment classifiers will further enhance these systems.

We summarize our contribution to AraSentEval 2026 Subtask 2 ([Ezzini et al., 2026](#)) as **ASBN-MT5**. From

the MA’AKS corpus ([Mughaus et al., 2026](#)), we report ~59 % flip success with zero empty predictions and a BLEU score of 30.79, competitive with the official test set. This offers an efficient and reproducible baseline for Arabic sentiment swap. Code and weights will be made public after acceptance.

Limitations

The negation-heuristic SFR is limited to the lexical sentiment alterations, mT5-small is smaller than the most current Arabic LLMs, a source-copy rate of 37.6 % suggests the generation is failing a lot, and the test-set targets are withheld by the organizers.

References

- Fahad Abdu, Raed Mughaus, Shadi Abudalfa, Mohammed Ahmed, and Ahmed Abdelali. 2025. [An empirical evaluation of Arabic text formality transfer: A comparative study](#). *Language Resources and Evaluation*.
- Fayez Aljomah, Lama Aldhafeeri, Mohammed Alfadel, Saad Alshahrani, Qaisar Abbas, and 1 others. 2025. [Enhancing Arabic sentiment analysis with pre-trained CAMELBERT: A case study on noisy texts](#). *Computers, Materials & Continua*, 84(3):5317–5335.
- Hamzah Almaqtari, Fang Zeng, and Abdulaziz Mohammed. 2024. [Enhancing Arabic sentiment analysis of consumer reviews: Machine learning and deep learning methods based on NLP](#). *Algorithms*, 17(11):495.
- Saad Ezzini, Paul Rayson, Shadi Abudalfa, Maram Alharbi, and Mo El-Haj. 2026. AraSentEval: A shared task on sentiment analysis and swapping in arabic. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with the Language Resources and Evaluation Conference (LREC 2026)*, Palma, Mallorca, Spain.
- Ibrahim Abu Farha and Walid Magdy. 2024. [Arabic sentiment analysis: Challenges and opportunities](#). *ACM Computing Surveys*.
- Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Mosen Alharthi, Bang An, and 1 others. 2024. [AceGPT, localizing large language models in Arabic](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 8139–8163, Mexico City, Mexico. Association for Computational Linguistics.
- Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. [Deep learning for text style transfer: A survey](#). *Computational Linguistics*, 48(1):155–205.
- Anis Koubaa, Adel Ammar, Lahouari Ghouti, Omer Nacar, and Serry Sibae. 2025. [ArabianGPT: An Arabic language adaptation of GPT-2](#). In *Generative AI*

- and Large Language Models: Opportunities, Challenges, and Applications, volume 1214 of *Studies in Computational Intelligence*. Springer, Cham.
- Yuning Lyu, Tongshuang Luo, Jinliang Shi, Todd Holton, and Honglak Lee. 2023. [Fine-grained text style transfer with diffusion-based language models](#). In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 65–74. Association for Computational Linguistics.
- Sarah Mankarious and Ayah Zirikly. 2025. [Style transfer as bias mitigation: Diffusion models for synthetic mental health text for Arabic](#). In *arXiv preprint arXiv:2601.14124*.
- Raed Mughaus, Shadi Abudalfa, Hamzah Luqman, Fahaad Abdu, Mohammed AlAli, Nawaf Al-Dowayan, and Ahmed Abdelali. 2026. [Ma’aks: manually-curated parallel dataset for Arabic text sentiment swap](#). *Language Resources and Evaluation*, 60(1):1.
- Sourabrata Mukherjee, Atul Kr. Ojha, Pruthwik Mujadia, and Ondrej Dusek. 2024. [Are large language models actually good at text style transfer?](#) In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Bangkok, Thailand. Association for Computational Linguistics.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.
- Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. [Style transfer through back-translation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.
- Neha Sengupta, Sunil Kumar Siddiqui, Basil Boregowda, and 1 others. 2023. [Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models](#). *arXiv preprint arXiv:2308.16149*.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. [Style transfer from non-parallel text by cross-alignment](#). In *Advances in Neural Information Processing Systems*, volume 30, pages 6830–6841.
- 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 (NAACL-HLT)*, pages 483–498, Online. Association for Computational Linguistics.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [BERTScore: Evaluating text generation with BERT](#). In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.
- Mohamed Zouidine and Mohamed Khalil. 2025. [Large language models for Arabic sentiment analysis and machine translation](#). *Engineering, Technology & Applied Science Research*, 15(2):20737–20742.

L3IA-Subtask 1 at AraSentEval Shared Task: Multi-Dialect Arabic Sentiment Classification via a Transformer-Based Approach

Mohamed M'haouach, Kaouthar Elyoussoufi, Hamza Alami⁵, Abdessamad Benlahbib

L3IA Laboratory, Faculty of Sciences Dhar El Mahraz,
Sidi Mohamed Ben Abdellah University, Fez, 30003, Morocco
{mohamed.mhaouach, kaouthar.elyoussoufi, hamza.alami⁵,
abdessamad.benlahbib}@usmba.ac.ma

Abstract

This paper presents our system and findings for AraSentEval 2026 Subtask 1 on Arabic Dialect Sentiment Analysis. We propose an automated sentiment classification system grounded in advanced Natural Language Processing (NLP) techniques. The proposed approach leverages pre-trained Transformer-based architectures to categorize textual inputs into three sentiment polarities: positive, negative, and neutral. Initially, a text normalization procedure is applied to unify the orthographic and graphical variations characteristic of the Arabic language. This process is further complemented by repetition reduction techniques, which aim to mitigate textual noise and enhance the overall consistency of the data. Subsequently, the data are adapted to the requirements of the pre-trained models to ensure coherent tokenization. The processed texts are then encoded into numerical representations that serve as inputs during training and evaluation. Finally, we conduct a comprehensive benchmarking study of five Transformer-based architectures to assess their effectiveness. The best-performing experimental setup yielded remarkable results on the AraSentEval 2026 benchmark, achieving a micro-F1 score of 75.96% on the official test set.

Keywords: Sentiment analysis, machine learning, Arabic dialects

1. Introduction

Sentiment analysis in Arabic dialects constitutes a major challenge in Natural Language Processing (NLP). This challenge arises in a context shaped by the rapid expansion of web applications and social media platforms, which continuously generate large volumes of reviews and user-generated comments across diverse domains (Matrane et al., 2023). Although these data represent a valuable resource for large-scale opinion and emotion analysis, their effective exploitation in the Arabic context remains difficult due to the coexistence of Modern Standard Arabic (MSA) and multiple country-specific dialects. Unlike MSA, which benefits from a relatively standardized linguistic system, Arabic dialects lack orthographic normalization and exhibit substantial lexical, syntactic, and morphological variability. Consequently, this linguistic heterogeneity limits the ability of computational models to learn truly generalizable representations and may undermine the robustness of sentiment predictions.

In recent years, increasing attention has been devoted to incorporating Arabic dialects into sentiment analysis within the field of Natural Language Processing. Nevertheless, the majority of existing studies remain primarily focused on Modern Standard Arabic (MSA) (Al-Harbi, 2019). As a consequence, models mainly trained on MSA often struggle to generalize to the wide range of dialectal varieties, which may undermine the reliability and robustness of predictions in real-world settings. In this context, the AraSentEval 2026 shared task

(Ezzini et al., 2026; Alharbi et al., 2025a,c,b) introduces a common experimental framework for sentiment analysis across multiple Arabic dialects. Subtask 1 addresses this challenge as a multi-class classification problem (positive, negative, neutral), requiring systems to effectively handle dialectal linguistic variability despite the scarcity of high-quality annotated resources and the inherent linguistic heterogeneity that characterizes these varieties.

In this paper, we present our system submitted to the AraSentEval 2026 shared task, which aims to advance sentiment classification techniques across Arabic dialects within the hospitality domain. To encode the input texts, we investigate the use of several pre-trained language models, namely BERT (Devlin et al., 2019), AraBERT (Antoun et al., 2020), MARBERT (Abdul-Mageed et al., 2021), CAMELBERT (Inoue et al., 2021), and DarjaBERT (Gaanoun et al., 2025). Each model is fine-tuned under identical experimental settings to ensure a fair and systematic comparison. This comparative study enables us to identify the most effective architecture for sentiment classification in dialect-rich Arabic data. Furthermore, it provides empirical insights into the robustness and generalization capacity of Transformer-based models when deployed in multi-dialectal environments.

This paper is organized as follows. Section 2 presents the proposed method and describes the overall approach adopted in this work. Section 3 reports the experimental setup and discusses the obtained results. Finally, Section 4 concludes the paper by summarizing the main findings.

2. Method

This section briefly outlines the methodology adopted for the development of an Arabic Dialect Sentiment Classification system. The proposed framework follows a structured pipeline, spanning from data preparation to the final classification stage, with the overall training procedure illustrated in Figure 1.

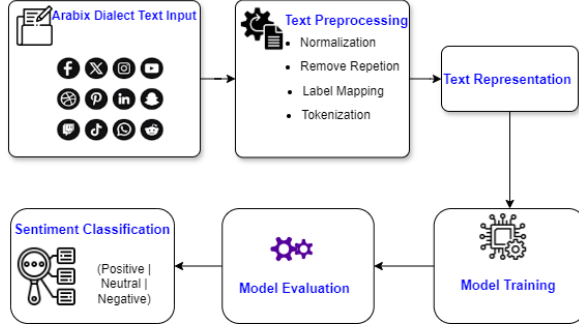


Figure 1: Flowchart of the Sentiment Classification System

2.1. Arabic Dialect Text Preprocessing

For the processing of Arabic dialectal texts, a lightweight preprocessing pipeline was employed to ensure compatibility with the underlying models while preserving the intrinsic linguistic information. An orthographic normalization step was introduced to standardize graphical variants and reduce formal variability that could otherwise amplify lexical sparsity. In addition, a label mapping procedure was performed, whereby the textual sentiment labels (“positive”, “negative”, “neutral”) were transformed into numerical identifiers (0, 1, 2), thus enabling their effective integration within supervised learning algorithms. Finally, each review was tokenized using the model-specific pre-trained tokenizer, thereby ensuring strict alignment between the input representation and the specifications of the corresponding pre-training phase.

2.2. Arabic Dialect Text Representation

To represent Arabic dialect texts, we employ contextualized embeddings generated by pre-trained Transformer architectures. Given an input sequence $X = (x_1, \dots, x_n)$, tokens are first mapped into dense embeddings combining token, positional, and segment information, and then processed through stacked self-attention layers to capture contextual and dialect-specific patterns. For classification, the final hidden state of the special token $[CLS]$, denoted as h_{CLS} , is used as a global representation of the text and fed into the downstream sentiment classifier.

2.3. Baseline Classifiers

In this study, we relied on five pre-trained Transformer-based architectures obtained from the HuggingFace Transformers library as our foundational models, namely: BERT, AraBERT, MARBERT, CAMELBERT, and DarijaBERT.

- **BERT**: A pre-trained Transformer-based language model developed by Google AI, designed to learn bidirectional contextual representations for various downstream NLP tasks
- **AraBERT**: A pre-trained Arabic Transformer model designed to achieve strong performance across multiple Arabic NLP benchmarks.
- **MARBERT**: is an Arabic Transformer-based language model pre-trained on large and diverse corpora to enable transfer learning for both Modern Standard Arabic and dialectal Arabic.
- **CAMELBERT**: is a suite of BERT-based models pre-trained on Arabic corpora, available in various sizes and configurations to support diverse NLP applications.
- **DarijaBERT**: is a suite of language models built upon the BERT architecture, specifically developed to address natural language processing tasks in the Moroccan Arabic dialect Darija.

3. Experimental Results

In this study, we conduct our experimental evaluation on AraSentEval 2026 – Subtask 1: Arabic Dialect Sentiment Analysis. All experiments were performed in the Google Colab environment¹ to ensure computational efficiency and reproducibility. Model implementation and fine-tuning were carried out using the Hugging Face Transformers library² (Wolf et al., 2020), while PyTorch³ was employed to handle the training procedure and performance evaluation. This experimental framework allows for a rigorous and systematic assessment of the proposed models under standardized and reproducible conditions.

3.1. Datasets

The experimental evaluation was conducted on the Multi-Dialect-Sent (MDS-3) dataset, a benchmark designed for sentiment analysis in Arabic dialects.

¹<https://colab.research.google.com/>

²<https://huggingface.co/docs/transformers/index>

³<https://pytorch.org/>

The corpus consists of 3,000 sentences annotated with three sentiment labels: positive, negative, and neutral. The data were collected from hotel reviews and subsequently translated and manually annotated by native speakers of the corresponding dialects. For the experiments, the dataset was divided into training (85%) and validation (15%) sets using label stratification to preserve the original class distribution. Figures 2 and 3 illustrate the distribution of sentiment categories and the distribution of Arabic dialects within the corpus.

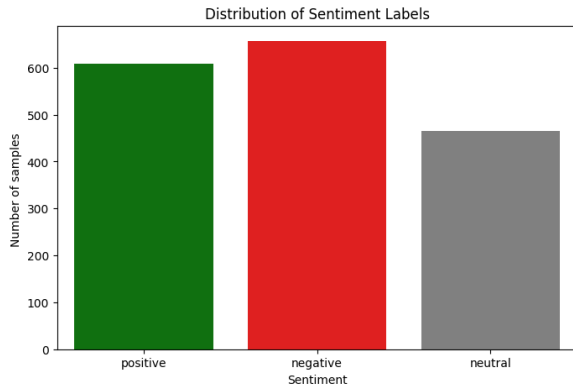


Figure 2: Distribution of Sentiment Labels

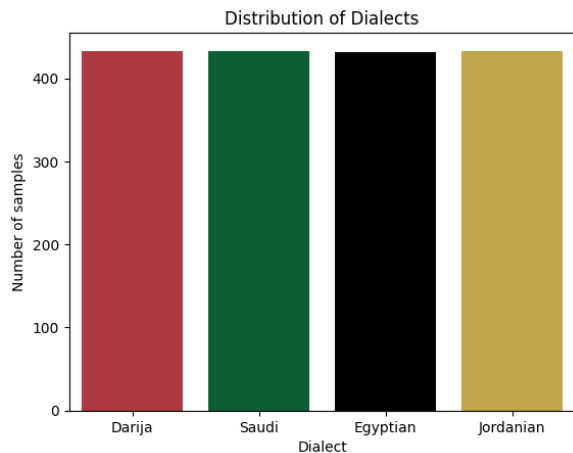


Figure 3: Distribution of Dialects

3.2. Experimental Settings

During the fine-tuning phase, all Transformer-based models were trained under a unified experimental configuration. The optimization process relied on the AdamW optimizer with a learning rate of 2×10^{-5} and a weight decay of $\lambda = 0.01$. The training procedure was conducted for 5 epochs using a batch size of 16. Model parameters were optimized using the Cross-Entropy loss function to handle the multi-class sentiment classification task.

3.3. Performance Evaluation

In this section, we present a comparative evaluation of the different models trained for Subtask 1. Their performance is assessed using standard classification metrics to measure their effectiveness in identifying sentiment polarity across multiple Arabic dialects.

In this experiment, we evaluate and compare several Transformer-based models for the task of Arabic dialect sentiment classification, as reported in Table 1 and illustrated in Figure 4. The results clearly indicate that MARBERT achieves the best overall performance, obtaining the highest Accuracy (95.10%) and F1-score (94.92%), which highlights its strong ability to model dialectal variations. AraBERT follows with competitive results (93.08% Accuracy and 92.54% F1-score), demonstrating balanced predictive performance. CAMELBERT and DarijaBERT achieve slightly lower yet comparable performances, with F1-scores of 90.16% and 90.45%, respectively. In contrast, BERT records the lowest performance (88.50% F1-score), likely due to its general-purpose pre-training not being specifically optimized for Arabic dialectal data.

Table 1: Performance models

Model	Accuracy	Precision	Recall	F1-score
BERT	88.76%	88.37%	88.69%	88.50%
AraBERT	93.08%	93.52%	92.12%	92.54%
MARBERT	95.10%	94.95%	94.92%	94.92%
CAMELBERT	90.49%	90.08%	90.25%	90.16%
DarijaBERT	90.78%	90.97%	90.32%	90.45%

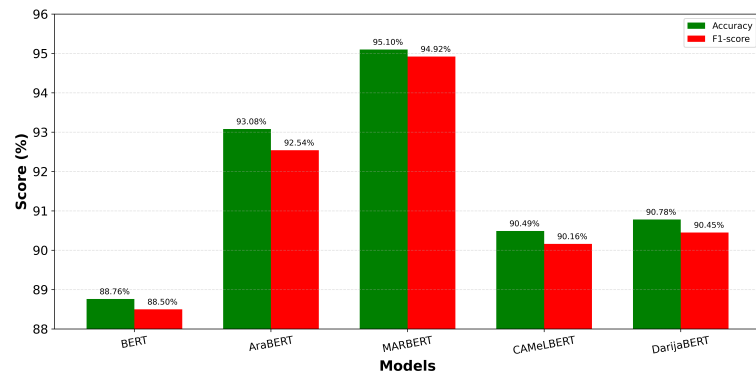


Figure 4: Comparative Performance of the Models

4. Conclusion

This work presented our system for AraSentEval 2026 Subtask 1, which addresses sentiment classification across diverse Arabic dialects. The proposed approach relies on a Transformer-based framework combined with a lightweight preprocessing pipeline, including orthographic normal-

ization and label mapping. Several pre-trained language models were evaluated in order to determine their effectiveness for multi-dialect sentiment analysis. Experimental results on the Multi-Dialect-Sent (MDS-3) dataset show that Transformer-based models can effectively capture sentiment polarity in dialect-rich Arabic texts. Among the evaluated architectures, MARBERT achieved the best performance, reaching an accuracy of 95.10% and a macro F1-score of 94.92%, highlighting the advantage of dialect-aware pre-trained models for handling linguistic variability in Arabic dialects.

5. References

- Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. [ARBERT & MARBERT: Deep bidirectional transformers for arabic](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), Volume 1: Long Papers*, pages 7088–7105. Association for Computational Linguistics.
- Omar Al-Harbi. 2019. Classifying sentiment of dialectal arabic reviews: A semi-supervised approach. *International Arab Journal of Information Technology*.
- Maram Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025a. Evaluating large language models on arabic dialect sentiment analysis. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing (RANLP)*.
- Maram I Alharbi, Salmane Chafik, Saad Ezzini, Ruslan Mitkov, Tharindu Ranasinghe, and Hansi Hettiarachchi. 2025b. Ahasis: Shared task on sentiment analysis for arabic dialects. In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 1–6.
- Maram I Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025c. Ador: Dataset for arabic dialects in hotel reviews: A human benchmark for sentiment analysis. In *Proceedings of the First Workshop on Advancing NLP for Low-Resource Languages*, pages 187–191.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT 2020), with a Shared Task on Offensive Language Detection*, pages 9–15. European Language Resources Association (ELRA).
- 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)*, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Saad Ezzini, Shadi Abudalfa, Maram Alharbi, Salmane Chafik, Samaher Alghamdi, Reem Alotaibi, Hamzah Luqman, Mo El-Haj, and Paul Rayson. 2026. AraSentEval: A shared task on sentiment analysis and swapping in arabic. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with the Language Resources and Evaluation Conference (LREC 2026)*, Palma, Mallorca, Spain.
- Kamel Gaanoun, Abdou Mohamed Naira, Anass Allak, and Imade Benelallam. 2025. Darijabert: A step forward in nlp for the written moroccan dialect. *International Journal of Data Science and Analytics*, 20(2):917–929.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop (WANLP 2021)*, pages 92–104, Kyiv, Ukraine (Online). Association for Computational Linguistics.
- Youssef Matrane et al. 2023. A systematic literature review of arabic dialect sentiment analysis. *Journal of King Saud University – Computer and Information Sciences*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Jamie Brew, Canwen Huang, Hug-gingFace Inc., et al. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

University of Tripoli at AraSentEval: Fine-Tuning MARBERTv2 and CAMELBERT for Multi-Dialect Arabic Sentiment Analysis

Amani Bahlul Sharif, Sarah Farag S. Hmeid, Abdusalam F.A. Nwesri

University of Tripoli

Am.Sharif@uot.edu.ly, Sa.Hamid@uot.edu.ly, a.nwesri@uot.edu.ly

Abstract

This paper presents our contribution to the AraSentEval 2026 shared task, specifically for Subtask 1: Arabic Dialect Sentiment Analysis, hosted at the OSACT7 workshop during LREC 2026. The task focuses on classifying the sentiment (positive, negative, neutral) of text written in four major Arabic dialects: Moroccan, Egyptian, Jordanian, and Saudi. We addressed this by fine-tuning several pre-trained language models, including MARBERTv2 and CAMELBERT, on the provided Multi-Dialect-Sent (MDS-3) dataset. Our best-performing system MARBERTv2, achieved a Macro F1-score of 84.29% on the official test set, securing fourth place among 13 participating teams. Our findings underscore the value of leveraging large pre-trained models tailored to dialectal Arabic for improved sentiment classification in this under-resourced domain.

Keywords: Natural Language Processing (NLP), Sentiment Analysis(SA), Arabic Dialects, Fine-tuning, MARBERTv2, CAMELBERT.

1. Introduction

The rapid growth of user-generated content on social media, review platforms, and online forums has led to an explosion of opinionated Arabic text, making sentiment analysis a key task for understanding public attitudes toward products, services, and social issues [Abo et al. \(2019\)](#). While sentiment analysis is now a mature field for English and other high-resource languages, work on Arabic remains relatively limited despite Arabic's status as one of the most widely used languages on the web [Oueslati et al. \(2020\)](#).

Arabic has its own challenges in sentiment analysis, given its complex morphology, word-formation processes, the fact that it's a right-to-left script, as well as the presence of Modern Standard Arabic (MSA) and its many dialects, which vary in vocabulary, spelling, and grammar [Katat et al. \(2024\)](#). These dialects have become very common in social media, which has its own set of complexities, such as the use of informal spelling, code-switching, the use of emojis, etc., as mentioned in [Shamsi and Abdallah \(2021\)](#). As a result, many techniques that work well for English or even MSA often degrade significantly when applied to noisy, dialectal Arabic text.

Over the past decade, research has explored a variety of approaches to Arabic sentiment analysis, ranging from lexicon-based and traditional machine learning methods to deep learning and transformer-based architectures [Katat et al. \(2024\)](#). Surveys consistently report that deep learning models and, more recently, pre-trained language models (e.g., AraBERT-like architectures) tend to outperform classic classifiers, especially when combined with suitable text preprocessing and rich

embeddings [Katat et al. \(2024\)](#). Nevertheless, progress is constrained by persistent obstacles such as the scarcity of high-quality labeled datasets, inconsistent evaluation protocols, and limited coverage of dialects and domains [Aliyu et al. \(2024\)](#).

Shared tasks and evaluation campaigns have a significant role to play in filling these gaps by providing standardized benchmarks and common metrics and through competitive leaderboards that allow for fair comparison and methodological innovation [Muaad et al. \(2025\)](#). The AraSentEval benchmark follows this line of work by offering a unified evaluation framework for Arabic sentiment analysis across multiple datasets and domains. It encourages researchers to tackle core challenges—including dialectal variation, domain shift, and noisy user-generated text—and to systematically compare traditional machine learning pipelines with modern transformer-based models.

This paper describes the system that was submitted to the AraSentEval competition, and it is hoped that the reader will find the information provided to be informative and helpful in understanding the effectiveness of transformer-based models in Arabic sentiment analysis, which is demonstrated through the fourth position achieved on the leaderboard.

2. Related Work

Previous research on Arabic sentiment analysis has largely focused on Modern Standard Arabic (MSA), while Arabic dialects have received comparatively limited attention [Matrane et al. \(2023\)](#). Recent surveys confirm that most available datasets and sentiment lexicons were designed for MSA, whereas dialectal resources remain relatively

small, domain-specific, or still under development [Aladeemy et al. \(2024\)](#); [Alharbi et al. \(2025b\)](#).

Early approaches to Arabic sentiment analysis relied heavily on lexicon-based methods, where sentiment polarity is inferred using predefined lists of annotated words. Notable resources such as the Arabic Senti-Lexicon and the Multi-domain Arabic Sentiment Corpus demonstrated that sentiment lexicons can significantly improve classification performance when combined with traditional classifiers [Al-Moslmi et al. \(2018\)](#); [Badaro et al. \(2014\)](#); [Assiri et al. \(2018\)](#); [Alsemaree et al. \(2024\)](#).

To address the limitations of lexicon-based approaches—particularly their reliance on manual resources and limited contextual understanding—researchers increasingly adopted supervised machine learning techniques. Over the past decade, algorithms such as Support Vector Machines (SVM) and Naive Bayes have consistently demonstrated strong performance in Arabic sentiment analysis, commonly using bag-of-words or TF-IDF features [Qader et al. \(2019\)](#); [enc \(2010\)](#); [Aladeemy et al. \(2024\)](#). Hybrid approaches that combine lexicon-based features with n-gram or embedding representations have shown further improvements in classification accuracy [Matrane et al. \(2023\)](#); [Jaber et al. \(2025\)](#). Additionally, semi-supervised and ensemble methods have been explored for low-resource dialectal datasets, achieving F1 scores approaching 89% on Algerian and mixed MSA/dialect Facebook data [Jaber et al. \(2025\)](#).

More recently, the field has shifted toward large pre-trained transformer models, which enable end-to-end fine-tuning rather than manual feature engineering. Models such as AraBERT and ArabicBERT have demonstrated significant improvements over traditional machine learning and earlier deep learning approaches on benchmark datasets including SemEval, ASTD, and ArSAS [Alduailej and Alothaim \(2022\)](#). Subsequent transformer models specifically designed for dialectal Arabic—such as MARBERT, CAMELBERT and QARiB—have further improved performance, particularly for noisy Twitter data and multi-dialect corpora, establishing strong state-of-the-art baselines for dialect sentiment analysis [Bourahouat et al. \(2024\)](#); [Alotaibi and Nadeem \(2025\)](#); [Mutawa and Sruthi \(2025\)](#).

Another important development is the CAMELBERT family of models developed by CAMEL Lab. These BERT-based encoders are pre-trained on large and diverse Arabic corpora covering MSA, dialectal Arabic (DA), Classical Arabic (CA), and mixed text varieties [Alotaibi and Nadeem \(2025\)](#); [Mutawa and Sruthi \(2025\)](#). Specialized variants such as CAMELBERT-DA-sentiment, which are fine-tuned on major Twitter datasets including

ASTD, ArSAS, and SemEval, have achieved strong F1 scores and often outperform multilingual models while remaining competitive with MARBERT on dialectal sentiment tasks [Alotaibi and Nadeem \(2025\)](#); [Alahmadi \(2025\)](#).

Several approaches were also evaluated in the AHaSiS 2025 shared task on Arabic dialect sentiment analysis ([Alharbi et al., 2025b](#)). The top-ranked system employed an ensemble of MARBERTv2, SaudiBERT, and DarijaBERT, achieving a micro-F1 score of 0.81 [Al-Khalifa \(2025\)](#). The second-ranked system used fine-tuning strategies for AraBERTv02, achieved a micro-F1 score of 0.7916 [Jaballah et al. \(2025\)](#). The third-ranked LBY team evaluated several transformer models—including MARBERTv2, AraBERT variants, QARiB, and DarijaBERT—and found that MARBERTv2 achieved the best performance, relying primarily on careful hyperparameter tuning rather than ensembling, with an F1 score of 0.79 ([Nwesri et al., 2025](#)).

Overall, current research indicates that Arabic-specific transformer models pre-trained on multi-dialect corpora, particularly MARBERT, AraBERT, QARiB, and CAMELBERT, represent the most effective family of models for Arabic and dialectal sentiment analysis and frequently serve as the foundation for state-of-the-art single or ensemble systems [Mansour et al. \(2025\)](#); [Mutawa and Sruthi \(2025\)](#).

3. Methodology

3.1. Task

Subtask 1 of AraSentEval 2026 focuses on Arabic Dialect Sentiment Analysis, a multi-class classification problem. The goal is to predict the sentiment polarity of text written in various Arabic dialects. Participants were provided with a training dataset containing sentences from four major dialects: Moroccan, Egyptian, Jordanian, and Saudi. Each sentence is labeled with one of three sentiment classes: positive, negative, or neutral. During the evaluation phase, participants received an unlabeled test set and were required to submit predicted labels for each instance. The official evaluation metric for this subtask is the Macro F1-score, which calculates the F1-score for each class independently and then takes their average. This metric was chosen by the organizers to ensure fair evaluation across all sentiment categories, regardless of any potential class imbalance in the test data [Ezzini et al. \(2026\)](#). The task challenges participants to develop models that can generalize across multiple dialects, handling the lexical, morphological, and syntactic variations that distinguish each dialect from both Modern Standard Arabic and from one another.

3.2. Dataset

The dataset consists of 1,731 hotel reviews written in four Arabic dialects: Saudi, Maghribi (Darija), Egyptian, and Jordanian. Text reviews are annotated as either “Positive”, “Negative”, or “Neutral”, and the specific dialect is included for each entry. The training set is well-distributed across these dialects; the Saudi, Darija, and Jordanian subsets each contain 152 positive, 166 negative, and 115 neutral reviews, while the Egyptian subset contains 153 positive, 159 negative, and 120 neutral reviews (Alharbi et al., 2025c).

The test set, released later during the evaluation phase, comprises 312 reviews (Alharbi et al., 2025a). Participants are required to predict the sentiment polarity of these reviews using their developed models. An additional column containing the predicted labels must be added to the test set prior to submission on the shared task platform, where automatic evaluation is conducted. Each participant is allowed a maximum of 10 submissions during the evaluation phase. Each team is allowed to show its best result on a public leaderboard, showcasing the performance of all participating teams.

3.3. Models

The benchmark for this task is the pre-trained BERT-based model, which was fine-tuned on Modern Standard Arabic (MSA) and various Arabic dialects. In our proposed method, the emphasis is on using and fine-tuning state-of-the-art models that have been developed to address the language complexities of both Modern Standard Arabic and the various Arabic dialects. For this purpose, the MARBERTv2 model, which is a strong language model developed to work on dialectal Arabic, and the models developed by the CAMEL-Lab particularly the "CAMEL-Lab/bert-base-arabic-camelbert-mix-sentiment", which have gained recognition for their strong performance on Natural Language Processing tasks in the Arabic language, have been used. They were chosen based on their effectiveness in identifying the sentiment expressed in the Saudi, Maghribi, Egyptian, and Jordanian dialects used in the dataset.

3.4. Evaluation Measure

The models were primarily evaluated using the F1-score and accuracy across the entire dataset, encompassing all four dialects simultaneously. This approach allowed for a direct comparison between MARBERTv2 and CAMELBERT to determine their overall performance on mixed dialectal Arabic.

4. Experiments: Models training

In our research, we decided to fine-tune two of the most advanced Arabic pre-trained models, namely MARBERTv2 and CAMELBERT, for the multi-dialectal sentiment analysis task. The fine-tuning was done using the Hugging Face’s Trainer API, while all the experiments were carried out on the Google Colab platform Bisong (2019).

It should be noted that the dataset was divided into 80% for training and 20% for evaluation. During the preprocessing stage, the dedicated tokenizer associated with each model was employed. The text was processed through tokenization, truncation, and padding, with the maximum sequence length set to 128 tokens. In addition, several hyperparameters were optimized to achieve the best performance, including the learning rate, batch size, weight decay, and warmup steps. The models were trained for a predefined number of epochs, with the evaluation conducted at the end of each epoch. The macro-averaged F1 score (F1-macro) was used as the primary evaluation metric throughout the experiments. The best performance was obtained using the following hyperparameter configuration: a learning rate of $1e-4$, batch size of 16, weight decay of $1e-4$, and warmup steps of 0.1. Under these settings, the models were trained for 8 epochs, with evaluation performed after each epoch.

4.1. Fine-tuning on the Original Dataset

Both MARBERTv2 and CAMELBERT models were fine-tuned using the original training dataset. Table 1 presents the best results achieved during the internal validation phase. Both models demonstrated strong performance, with improvements resulting from careful hyperparameter tuning. Among the two models, MARBERTv2 outperformed CAMELBERT, achieving an F1-macro score of 97% on the validation set. The CAMELBERT model followed closely, obtaining a competitive F1-macro score of 96.

4.2. Fine-tuning on the Preprocessed Dataset

Social media text often contains various forms of noise, including emojis, symbols, and elongated characters used to express emotions. In addition, Arabic dialectal text is characterized by inconsistent writing styles, where the same word may appear in multiple orthographic forms due to informal spelling. This issue is commonly addressed through text normalization, where different letter variants are unified into a single standardized form.

In this experiment, the training dataset was pre-processed prior to model fine-tuning by removing

all hyperlinks, strings beginning with the @ symbol, diacritics, repeated characters, normalizing specific Arabic letters by replacing أ or إ with ا ; ة with ه ; and ى with ي , and finally removing any extra spaces within the text.

Table 1 presents the results obtained by fine-tuning the models using both the original dataset and the preprocessed dataset.

The results show that training on the original dataset consistently outperformed training on the preprocessed dataset across all evaluation metrics. For MARBERTv2, training on the original data achieved slightly higher performance, with an accuracy of 0.964 and F1 score of 0.965, compared with 0.960 accuracy and 0.958 F1 on the preprocessed data, along with similar improvements in precision (0.963 vs. 0.953) and recall (0.964 vs. 0.955). A larger performance drop was observed for CAMELBERT, where the model achieved 0.955 accuracy and 0.960 F1 on the original dataset but only 0.936 accuracy and 0.932 F1 after preprocessing, with precision and recall also decreasing notably. While MARBERTv2 consistently outperformed CAMELBERT on the preprocessed dataset, the performance gap between the two models became smaller when trained on the original dataset. Overall, these findings suggest that aggressive preprocessing and text normalization may remove useful linguistic cues from Arabic social media text, whereas transformer-based models can effectively handle noisy and informal text, benefiting more from the original unprocessed data.

5. Results on the testset

Both trained models were run on the released unlabeled dataset. Table 2 shows results obtained using the trained models. Out of the 8 submissions we made during the test phase, the MARBERTv2 result was submitted to the leaderboard under the name of "University of Tripoli". The result placed us at the forth position among 13 participating teams.

6. Discussion

The experimental results demonstrate the effectiveness of fine-tuning large pre-trained Arabic language models for the task of multi-dialectal sentiment analysis. Both MARBERTv2 and CAMELBERT achieved strong performance during the validation phase, confirming the capability of transformer-based architectures to capture sentiment information across diverse Arabic dialects. The results also highlight the advantage of domain-specific pre-trained models that have been trained on large-scale Arabic corpora containing dialectal and social media content.

A key observation from the experiments is that training on the original dataset consistently produced better results than training on the preprocessed dataset. While preprocessing techniques such as removing diacritics, normalizing letters, and eliminating repeated characters are commonly used in traditional natural language processing pipelines, the results indicate that these operations may inadvertently remove useful contextual or stylistic information embedded in social media text. Elements such as character elongation and informal orthographic variations often carry implicit sentiment or emphasis, which transformer-based models can learn to interpret effectively. Consequently, aggressive normalization may reduce the richness of the input data and negatively affect model performance.

The impact of preprocessing was particularly evident in the CAMELBERT model, which experienced a more substantial performance drop when trained on the preprocessed dataset compared with MARBERTv2. This behavior may be attributed to differences in the pre-training corpora and tokenization strategies used by the two models. MARBERTv2, which is specifically optimized for Arabic social media text, appears to be more robust to textual variations and noise, allowing it to maintain relatively stable performance even after preprocessing. In contrast, the CAMELBERT model appears to rely more heavily on the original lexical patterns present in the raw data.

The evaluation on the unseen test set further confirms the effectiveness of the trained models. MARBERTv2 achieved an F1 score of 0.843, outperforming CAMELBERT, which obtained an F1 score of 0.824. These results are consistent with the validation experiments and demonstrate that MARBERTv2 provides better generalization for the multi-dialectal sentiment analysis task. The submitted MARBERTv2 model ranked fourth among 13 participating teams on the official leaderboard, indicating competitive performance within the shared task.

Overall, the findings suggest that modern transformer-based language models can effectively handle the inherent noise and variability present in Arabic social media text, reducing the need for extensive preprocessing. Instead, preserving the original textual structure may allow the models to leverage subtle linguistic cues that contribute to sentiment detection.

7. Conclusion

This study evaluated the effectiveness of fine-tuning two state-of-the-art Arabic pre-trained language models, MARBERTv2 and CAMELBERT, for multi-dialectal sentiment analysis using the Hugging

Model	Original Dataset				Preprocessed Dataset			
	Acc.	F1	P	R	Acc.	F1	P	R
MARBERTv2	0.964	0.965	0.963	0.964	0.960	0.958	0.953	0.955
CAMELBERT	0.955	0.960	0.958	0.953	0.936	0.932	0.934	0.934

Table 1: Results of fine-tuning models on the preprocessed and the original training dataset.

Model	Acc.	F1	P	R
MARBERTv2	0.843	0.843	0.843	0.843
CAMELBERT	0.824	0.824	0.824	0.824

Table 2: Results of running trained models on the unlabeled testset.

Face Trainer framework and the macro-averaged F1 score as the main evaluation metric. Experimental results showed that both models achieved strong performance, with MARBERTv2 consistently outperforming CAMELBERT, reaching the best validation performance with an F1-macro score of 0.965 when trained on the original dataset. The study also compared training on the original dataset and a normalized preprocessed version, finding that the original dataset yielded better results, suggesting that aggressive preprocessing may remove useful linguistic cues from Arabic social media text. Evaluation on the official test set further confirmed these findings, where MARBERTv2 achieved an F1 score of 0.843, ranking fourth among 13 participating teams in the shared task leaderboard. These results demonstrate the effectiveness of transformer-based Arabic language models for handling dialectal variation in sentiment analysis, while future work may explore ensemble approaches, data augmentation, dialect-aware preprocessing, and larger language models to further improve performance.

8. References

2010. [Encyclopedia of machine learning](#).

Mohamed Elhag Mohamed Abo, Ram Gopal Raj, and Atika Qazi. 2019. [A review on arabic sentiment analysis: State-of-the-art, taxonomy and open research challenges](#). *IEEE Access*, 7:162008–162024.

Hend Al-Khalifa. 2025. [iwan-nlp at ahasis 2025: A stacked ensemble of arabic transformers for sentiment analysis on arabic dialects in the hospitality domain](#). In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 7–13, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Tareq Al-Moslmi, M. Albared, Adel Al-Shabi, N. Omar, and S. Abdullah. 2018. [Arabic senti-](#)

[lexicon: Constructing publicly available language resources for arabic sentiment analysis](#). *Journal of Information Science*, 44:345 – 362.

Amani A. Aladeemy, Ali Alzahrani, Mohammad H. Algarni, S. N. Alsubari, Theyazn H. H. Aldhyani, Sachin N. Deshmukh, O. I. Khalaf, Wing-Keung Wong, and Sameer Aqburi. 2024. [Advancements and challenges in arabic sentiment analysis: A decade of methodologies, applications, and resource development](#). *Heliyon*, 10.

D. Alahmadi. 2025. [Human versus ai: A comparative study of zero-shot llms and transformer models against human annotations for arabic sentiment analysis](#). *International Journal of Advanced Computer Science and Applications*.

Alhanouf Alduailej and Abdulrahman Alothaim. 2022. [AraxInet: pre-trained language model for sentiment analysis of arabic](#). *Journal of Big Data*, 9.

Maram Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025a. Evaluating large language models on arabic dialect sentiment analysis. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing (RANLP)*.

Maram I. Alharbi, Salmane Chafik, Saad Ezzini, Ruslan Mitkov, Tharindu Ranasinghe, and Hansi Hettiarachchi. 2025b. [AHasIS: Shared task on sentiment analysis for Arabic dialects](#). In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 1–6, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Maram I Alharbi, Saad Ezzini, Hansi Hettiarachchi, Tharindu Ranasinghe, and Ruslan Mitkov. 2025c. Ador: Dataset for arabic dialects in hotel reviews: A human benchmark for sentiment analysis. In *Proceedings of the First Workshop on Advancing NLP for Low-Resource Languages*, pages 187–191.

Yusuf Aliyu, A. Sarlan, Kamaluddeen Usman Danyaro, A. S. B. A. Rahman, and Mujahed Abdulahi. 2024. [Sentiment analysis in low-resource settings: A comprehensive review of approaches, languages, and data sources](#). *IEEE Access*, 12:66883–66909.

- Alanoud Alotaibi and Farrukh Nadeem. 2025. [An unsupervised integrated framework for arabic aspect-based sentiment analysis and abstractive text summarization of traffic services using transformer models](#). *Smart Cities*.
- Ohud Alsemaree, A. Alam, S. Gill, and Steve Uhlig. 2024. [An analysis of customer perception using lexicon-based sentiment analysis of arabic texts framework](#). *Heliyon*, 10.
- Adel M. A. Assiri, Ahmed Z. Emam, and Hmood Al-Dossari. 2018. [Towards enhancement of a lexicon-based approach for saudi dialect sentiment analysis](#). *Journal of Information Science*, 44:184 – 202.
- Gilbert Badaro, Ramy Baly, Hazem Hajj, Nizar Habash, and Wassim El-Hajj. 2014. [A large scale arabic sentiment lexicon for arabic opinion mining](#). In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 165–173. Association for Computational Linguistics.
- Ekaba Bisong. 2019. [Building Machine Learning and Deep Learning Models on Google Cloud Platform](#). Apress, Berkeley, CA.
- Ghizlane Bourahouat, Manar Abourezq, and N. Daoudi. 2024. [Improvement of moroccan dialect sentiment analysis using arabic bert-based models](#). *Journal of Computer Science*.
- Saad Ezzini, Shadi Abudalfa, Maram Alharbi, Salmane Chafik, Samaher Alghamdi, Reem Alotaibi, Hamzah Luqman, Mo El-Haj, and Paul Rayson. 2026. AraSentEval: A shared task on sentiment analysis and swapping in arabic. In *Proceedings of the 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7), co-located with the Language Resources and Evaluation Conference (LREC 2026)*, Palma, Mallorca, Spain.
- Mustapha Jaballah, Dhaou Ghoul, and Ammar Mars. 2025. [Fine-tuning arabert model for arabic sentiment detection](#). In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 14–23, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.
- Areej Jaber, Israa Bahati, and Paloma Martínez. 2025. [Leveraging pre-trained embeddings in an ensemble machine learning approach for arabic sentiment analysis](#). *Frontiers in Artificial Intelligence*, 8.
- Souha Al Katat, Chamseddine Zaki, Hussein Hazimeh, Ibrahim El Bitar, Rafael Angarita, and Lionel Trojman. 2024. [Natural language processing for arabic sentiment analysis: A systematic literature review](#). *IEEE Transactions on Big Data*, 10:576–594.
- Omar Mansour, Eman Aboeela, Remon Talaat, and Mahmoud Bustami. 2025. [Transformer-based ensemble model for dialectal arabic sentiment classification](#). *PeerJ Computer Science*, 11.
- Yassir Matrane, F. Benabbou, and N. Sael. 2023. [A systematic literature review of arabic dialect sentiment analysis](#). *J. King Saud Univ. Comput. Inf. Sci.*, 35:101570.
- Abdullah Y. Muaad, Md Belal Bin Heyat, F. Akhtar, Usman Naseem, Wadea R Naji, Suresha Mallappa, and H. J. 2025. [Artificial intelligence for text analysis in the arabic and related middle eastern languages: Progress, trends, and future recommendations](#). *Int. J. Intell. Syst.*, 2025.
- A. Mutawa and S. Sruthi. 2025. [A comparative evaluation of transformers and deep learning models for arabic meter classification](#). *Applied Sciences*.
- Abdusalam F. Ahmad Nwesri, Nabila Almabrouk S. Shinbir, and Amani Bahlul Sharif. 2025. [Sentiment analysis on arabic dialects: A multi-dialect benchmark](#). In *Proceedings of the Shared Task on Sentiment Analysis for Arabic Dialects*, pages 86–91, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.
- Oumaima Oueslati, E. Cambria, M. Hajhmida, and Habib Ounelli. 2020. [A review of sentiment analysis research in arabic language](#). *ArXiv*, abs/2005.12240.
- Raheel Qader, François Portet, and Cyril Labbé. 2019. Semi-supervised neural text generation by joint learning of natural language generation and natural language understanding models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 552–562.
- Arwa A. Al Shamsi and Sherief Abdallah. 2021. [Text mining techniques for sentiment analysis of arabic dialects: Literature review](#). *Advances in Science, Technology and Engineering Systems Journal*, 6:1012–1023.

Author Index

- abdelaziz, chaima, 278
Abu Adla, Yasmine, 115
Abudalfa, Shadi, 256
AbuOdeh, Muhammed, 94
Abusaleh, Ali, 262
Ahmed, Samar Essa, 88
Al-Ghamdi, Sharefah Ahmed, 151
Al Hallay, Rahaf Saeed, 88
Al Jallad, Khloud, 40, 76
Al-Khalifa, Hend, 81, 141, 151
Al Moubayed, Noura, 25
Al-rasheed, Raghad, 220
Al Tabash, Kholood, 252
Al Wazrah, Asma Ali, 220
Alabrash, Buthaina, 115
Alami, Hamza, 274, 284, 292
Alamr, Meshal Abdullah, 225
Alangari, Nourah, 141
Alasmary, Faris, 252
Albahlal, Manal M., 141
Albaiti, Joud A., 14
Albalkhi, Eman M., 14
Albaqawi, Mona H., 14
Aldahlawi, Abdullah, 225
Aldarmaki, Hanan, 220
Alenezi, Reem, 106
Alfattni, Ghada, 157
Alfear, Noof Abdullah, 151
Alharbi, Abdullah I., 220
Alharbi, Maram I., 256
Alharthi, Naif Saad, 252
Alhazmi, Hend Hamed, 151
AlHumoud, Sarah, 35
Almasoud, Ameera Masoud, 141, 151
Almatham, Rawan, 220
Almazrui, Amal, 220
Alosaimy, Abdulrahman, 220
Alotaibi, Reem, 256
Alqaeri, Hassan Rshed, 225
Alqahtani, Sawsan, 220
Alqifari, Reem Fahad, 141, 151
alrabghi, ahmad, 88
Alshammari, Waad, 220
Alshehri, Abdulrahman Saeed, 220
ALSmadi, Mohammad, 199
Alsuhaimeh, Shaden, 88
Altamimi, Afrah Abdulaziz, 220
Alturki, Rand Abdullah, 185
Alzubaidi, Wareef, 234
Andriyanova-Almaamary, Nina A., 165
Asiri, Ali, 59
Assar, Mohamed, 220
Balah, Batool, 76
Barakat, Youssef, 35
Barmandah, Hassan, 234
Belabbaci, Amel, 49
BELBACHIR, Faiza, 278
Ben Amor, Ghada, 179
ben arbia, mariem, 179
Benlahbib, Abdessamad, 274, 284, 292
Bichi, Abdulkadir Shehu, 288
Bouchekef, Abdessalam, 191
Bouchekef, Chahinez, 204
Bougares, Fethi, 131
Bounnit, Maria, 151
Chafik, Salmane, 256
Cheragui, Mohamed Amine, 66
Cherroun, Hadda, 49
Chowdhury, Nahian, 247
Dahimi, Khaoula, 49
Dahou, Abdelhalim Hafedh, 66
Dekmak, Fatima, 115
Eid, Yomna, 269
El-Haj, Mo, 256
El-Hajj, Wassim, 115
Elbasiony, Reda M., 229
Elbassuoni, Shady, 115
ElDamhougy, Mohab, 269
Eldin, Fatimah Emad, 234
Elhosiny, Israa, 76
Elleuch, Haroun, 131
Elshehy, Omar, 40
Eltanbouly, Somaya, 191
Elyoussofi, Kaouthar, 274, 284, 292

Emad, MOHAMED, 191
 Esmaeil, Nada Adel, 229
 Ezzini, Saad, 256

 Faheem, Mohamed T., 229
 Farouk Zakaria Elshabrawy, Ahmed, 94

 Gaben, Shahd, 191
 Ghaly, Mohammed, 191
 Ghannam, Ahmad, 252
 Ghezaiel, Nadia, 151
 Ghouti, Lahouari, 252

 Habash, Nizar, 94
 Hajj, Hazem, 115
 Hamdy, Yumna, 269
 Hamod, Aya, 88
 Haouhat, Abdelhamid, 49
 Hefny Abdelkader, Abdelrahman Hamada, 1
 Hmeid, Sarah Farag S., 296
 Horaira, Muhammad Abu, 247
 Huang, Baorong, 59
 Hussein, Ensaf, 269

 Ibrahim, Mohamed, 35
 Inoue, Go, 94

 Justanieah, Hala, 213
 Justanieah, Hanan, 213

 Kurdi, Ghader, 213
 kurdi, ghader, 157

 Lopedoto, Enrico, 14
 Luqman, Hamzah, 256

 Makki, Abdullah, 35
 Marew, Rufael, 220
 Mehler, Alexander, 262
 M'haouach, Mohamed, 274, 284, 292
 Mohammed, amine, 204
 Mubarak, Nour Aldin Al, 25

 Nacar, Omer, 40, 76, 88, 234
 Nwesri, Abdusalam F. Ahmad, 296

 Qandos, Nour, 88

 Rashwani, Samer, 191
 Rayson, Paul, 256

 Saadaoui, KahinaHouda, 278
 Sadah, Shouq, 252
 Said Lhadj, Lynda, 278
 saidi, youssef, 131

 Salman, Ayed Atallah, 106
 Samy, Nour, 35
 Sbahi, Heba, 191
 Shaban, Khaled, 115
 Sharif, Amani Bahlul, 296
 Sibae, Serry, 76
 Sidaoui, Hicham Ghazi, 209

 Trigui, Omar, 179

 Verma, Bhuvanesh, 262

 Wu, Stephen, 174

 Yassine, Sarah, 288
 Yousfi, Zineb, 76
 Yousra El-Ghawi, Yousra, 76

 Zaytoon, Mohamed, 40
 Zwawi, Alla, 174