ArabicNLP 2025

The Third Arabic Natural Language Processing Conference

Volume 2. Proceedings of the ArabicNLP 2025 Shared Tasks

November 8-9, 2025

The ArabicNLP organizers gratefully acknowledge the support from the following sponsors.

Platinum

HUMAIN Google

Gold

مختبر كامل
CAMeL Lab

Clinical
AI Lab

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-356-2

Introduction

Welcome to the proceedings of the Shared Tasks at the **Third Arabic Natural Language Processing Conference (ArabicNLP 2025)**, co-located with **EMNLP 2025** in Suzhou, China, November 8–9, 2025. This volume represents a major achievement in collaborative Arabic NLP research, bringing together a total of **138 papers from 11 shared tasks** — including **11 overview papers** and **127 system description papers** from participating teams worldwide. This substantial collection reflects the growing vitality and maturity of the Arabic NLP research community and showcases the field's expansion into increasingly diverse and sophisticated challenges.
This collection highlights the community's growing interest toward evaluating, aligning, and extending these models for Arabic — across text, speech, and multimodal domains, as well as culturally grounded and ethically sensitive applications. This year marks a record milestone for ArabicNLP, featuring the largest number of shared tasks to date — **11 in total**. The process began with **26 submitted shared task proposals**, from which **11 shared tasks were accepted**, including **6 formed through successful mergers**. This outcome underscores the community's strong spirit of collaboration and its collective effort to design impactful, high-quality shared tasks.

Organization and Participation of the Shared Tasks

The 11 shared tasks at ArabicNLP 2025 are organized into three thematic tracks, each addressing critical needs and emerging priorities in Arabic language technology. **Together, they represent the community's most comprehensive shared task effort to date, covering speech, multimodal processing, text quality, and cultural and ethical evaluation in the era of large language models (LLMs).**

Track 1: Speech and Multimodal Processing This track features four shared tasks that advance Arabic language processing beyond traditional text:

- **ImageEval (Arabic Image Captioning)** — 8 papers
- **Iqra'Eval (Qur'anic Pronunciation Assessment)** — 5 papers
- **NADI 2025 (Multidialectal Arabic Speech Processing)** — 7 papers
- **MAHED 2025 (Multimodal Detection of Hope and Hate Emotions in Arabic Content)** — 23 papers

  These tasks address the pressing need for technologies that can process Arabic across different modalities and dialectal variations.

Track 2: Text Quality and Generation Assessment This track comprises four shared tasks focused on evaluating and enhancing Arabic text quality:

- **AraGenEval (Arabic Authorship Style Transfer and AI-Generated Text Detection)** — 16 papers
- **TAQEEM 2025 (Arabic Quality Evaluation of Essays in Multi-dimensions)** — 4 papers
- **BAREC 2025 (Arabic Readability Assessment)** — 17 papers
- **AraHealthQA 2025 (Comprehensive Arabic Health Question Answering)** — 15 papers

  These tasks tackle essential challenges in automated assessment, generation evaluation, and domain-specific question answering.

Track 3: Cultural and Ethical Evaluation of LLMs for Arabic This track introduces three groundbreaking shared tasks designed to evaluate large language models' understanding of Arabic culture and Islamic knowledge:

- **IslamicEval (Capturing LLMs' Hallucination in Islamic Content)** — 7 papers
- **PalmX 2025 (Benchmarking LLMs on Arabic and Islamic Culture)** — 9 papers
- **QIAS 2025 (Islamic Inheritance Reasoning and Knowledge Assessment)** — 16 papers

  These tasks represent a critical step toward ensuring that AI systems appropriately and accurately represent the cultural and religious contexts of Arabic-speaking communities.

Community Impact and Participation

The remarkable response to this year's shared tasks — with **127 system submissions across the 11 tasks** — demonstrates the Arabic NLP community's continued growth and dynamism. Participating teams represent a diverse range of institutions across multiple continents, including universities, research centers, and industry partners, reflecting both the global interest in Arabic NLP and the real-world relevance of these challenges.

The breadth of methodological approaches presented in these proceedings is particularly noteworthy, spanning traditional machine learning techniques, state-of-the-art transformer models, retrieval-augmented generation systems, and multimodal architectures. This methodological diversity not only reflects the field's technical maturity but also highlights researchers' creativity in addressing the unique challenges posed by Arabic language processing.

The breadth of methodological approaches presented in these proceedings is particularly noteworthy, spanning traditional machine learning techniques, state-of-the-art transformer models, retrieval-augmented generation systems, and multimodal architectures. This methodological diversity not only reflects the field's technical maturity but also highlights researchers' creativity in addressing the unique challenges posed by Arabic language processing.

Acknowledgments

We extend our sincere gratitude to the dedicated organizers of each shared task. Their expertise in task design, dataset development, evaluation framework creation, and ongoing participant support has been fundamental to the success of these shared tasks. The quality and impact of this volume are a direct reflection of their commitment and hard work.

We thank all participating teams whose innovative research pushes the boundaries of Arabic NLP. Their contributions address not only technical challenges but also important societal needs in education, healthcare, cultural preservation, content moderation, and ethical AI development.

We also acknowledge the program committee members, reviewers, and the ArabicNLP 2025 organizing committee for their invaluable support throughout the shared task cycle.

Finally, we gratefully acknowledge our platinum and gold sponsors whose support has made this conference and these proceedings possible.

We hope this volume serves as both a comprehensive record of the current state of Arabic NLP research and a foundation for future innovations that will continue to advance language technologies for Arabic-speaking communities worldwide.

Wajdi Zaghouani, Northwestern University in Qatar, Qatar.
Sakhar Alkhereyf, Humain, Saudi Arabia.
Shared Tasks Chairs

Website of the shared Tasks: https://arabicnlp2025.sigarab.org/shared-tasks

Organizing Committee

General Chair

Kareem Darwish, Qatar Computing Research Institute, Qatar

Program Chairs

Ahmed Ali, Humain, KSA
Ibrahim Abu Farha, Alsun AI, UK
Samia Touileb, University of Bergen, Norway
Imed Zitouni, Google, USA

Publication Chairs

Ahmed Abdelali, Humain, KSA
Sharefah Al-Ghamdi, King Saud University, KSA

Shared Tasks Chair

Sakhar Alkhereyf, Humain, KSA
Wajdi Zaghouani, Northwestern University in Qatar, Qatar

Publicity Chairs

Salam Khalifa, Stony Brook University, USA
Badr AlKhamissi, EPFL, Switzerland
Rawan Almatham, King Salman Global Academy for Arabic Language, Saudi Arabia

Scholarships and Awards Chairs

Injy Hamed, New York University Abu Dhabi, UAE
Zaid Alyafeai, KAUST, KSA

Sponsorship Chairs

Areeb Alowisheq, Humain, KSA
Imed Zitouni, Google, USA

Social Chairs

Go Inoue, MBZUAI, UAE
Khalil Mrini, TikTok, USA
Waad Alshammari, King Salman Global Academy for Arabic Language, Saudi Arabia

Program Committee

Reviewers

Abdelkader El Mahdaouy, Mohammed VI Polytechnic University
Abdellah El Mekki, University of British Columbia
AbdelRahim A. Elmadany, University of British Columbia
Abdelrhman Ahmed Yousry Elnainay, Alexandria University
Abdulaziz Alhamadani
Abdulkareem Alsudais, Prince Sattam bin Abdulaziz University
Abdulmohsen Al-Thubaity, Humain
Abdurahman Khalifa AAlAbdulsalam, Sultan Qaboos University
Abed Qaddoumi, State University of New York at Stony Brook
Abul Hasnat
Ahamed Rameez Mohamed Nizzad, British College of Applied Studies
Ahmad M Mustafa, Jordan University of Science and Technology
Ahmed Abdelali, Humain
Ahmed Taha, Whiterabbit.AI
Ahmed Wasfy
Ahmed Cherif Mazari, University of Médéa
Ahmed Oumar El-Shangiti, Mohamed Bin Zayed University of Artificial Intelligence
Alaa Aljabari, Birzeit University
Alexis Nasr, Aix Marseille University
Ali Al-Laith
Ali S. Al-Zawqari
Almoataz B. Al-Said
Aloulou Chafik, Univeristy of Sfax
Amel Muminovic, International Balkan University
Amir Hussein
Amr El-Gendy
Amr Keleg, University of Edinburgh, University of Edinburgh
Ann Bies, Linguistic Data Consortium, University of Pennsylvania
Ashraf Elnagar, Google
Ashwag Alasmari, King Khaled University
Attia Nehar
Badr M. Abdullah
Baraa Hikal
Bashar Alhafni, Mohamed bin Zayed University of Artificial Intelligence
Bashar Talafha, University of British Columbia
Caroline Sabty, German International University
Chaima Ben Rabah, weill cornell Medicine
Claudia Borg, University of Malta
David Corney, Full Fact
David M. Palfreyman, United Arab Emirates University
Duygu Altinok
El Moatez Billah Nagoudi, University of British Columbia
Elisa Gugliotta, CNR-Istituto di Linguistica Computazionale A. Zampolli"
Elsayed Issa, Purdue University
Enas Albasiri, NVIDIA
Eyob Nigussie Alemu, Addis Ababa University

Fadhl Eryani, Eberhard-Karls-Universität Tübingen
Fadi Zaraket, Arab Center for Research and Policy Studies and American University of Beirut
Fatima Haouari, University of Sheffield
Fethi Bougares, elyadata
Firoj Alam, Qatar Computing Research Institute
Ghassan Mourad, Lebanese University
Go Inoue, Mohamed bin Zayed University of Artificial Intelligence
Hadda Cherroun, Université Amar Telidji
Hamzah Luqman, King Fahad University of Petroleum and Minerals
Hoda Zaiton, Pharos University in Alexandria
Hossam Ahmed, Leiden University
Houda Bouamor, Carnegie Mellon University
Ibrahim Bounhas
Injy Hamed, Mohamed bin Zayed University of Artificial Intelligence
Irfan Ahmad, King Fahad University of Petroleum and Minerals
Ismail Berrada, Mohammed VI Polytechnic University
Kamel Gaanoun, Institut National de Statistiques et d'Economie Appliquées
Kedir Yassin Hussen
Khalil Hennara
Khloud Al Jallad, HIAST
Kurt Micallef, University of Malta
Maged Al-shaibani, SDAIA-KFUPM Joint Research Center for Artificial Intelligence
Majd Hawasly
Malik H. Altakrori, IBM TJ Watson Research Center
Marwan Torki, Alexandria University
Mayar Nassar, Ain Shams University
Minh Ngoc Ta
Mohamed Lichouri, Université des Sciences et de la Technologie Houari Boumediène
Mohamed Nabih, Fondazione Bruno Kessler
Mohamed Bayan Kmainasi, University of Qatar
Mohamed Motasim Hamed
Mohammed Attia, Google
Mohammed Salah Al-Radhi, Budapest University of Technology and Economics
Mona Abdelazim, Ain Shams University
Mouath Abu Daoud
Moustafa Wassel
Muhammad Shakeel, Honda Research Institution Japan Co., Ltd.
Muhammed AbuOdeh, New York University, Abu Dhabi
Mustafa Jarrar, Birzeit University
Nada Ghneim
Nada Sharaf, The German International University
Nizar Habash, New York University Abu Dhabi
Omar Trigui, Institut Supérieur de Gestion de Sousse
Omer Goldman, Bar Ilan University
Omer Nacar
Pagon Gatchalee
Panigrahi Srikanth
Pavel Denisov, Fraunhofer IAIS and University of Stuttgart
Peter Sullivan, University of British Columbia
Petr Zemánek, Charles University Prague
Preslav Nakov, Mohamed bin Zayed University of Artificial Intelligence

Rania Al-Sabbagh
Rania Azad M. San Ahmed, Sulaimani Polytechnic University, Sulaymaniyah, Iraq
Sadam Al-Azani, King Fahad University of Petroleum and Minerals
Saeed Ahmadnia, University of Illinois at Chicago
Saied Alshahrani, University of Bisha
Sakhar Alkhereyf, Humain
Salam Albatarni
Salam Khalifa, New York University and State University of New York, Stony Brook
Salima Harrat
Salima Mdhaffar, Université d'Avignon
Samhaa R. El-Beltagy
Sanaa Kaddoura
Seid Muhie Yimam, Universität Hamburg
Serry Sibaee, Prince Sultan University
Shah Nawaz, Johannes Kepler Universität Linz
Sharif Ahmed, University of Central Arkansas
Simran Tiwari, Mendel Health Inc
Slimane Bellaouar
Sohaila Eltanbouly, University of Qatar
Sultan Alrowili, IBM Research
Suveyda Yeniterzi, GenAIus Technologies
Taha Zerrouki
Tamer Elsayed, Qatar University
Usman Nawaz, University of Palermo, Italy
Vincent Koc, Comet ML, The University of Queensland and Massachusetts Institute of Technology
Violetta Cavalli-Sforza
Waad Thuwaini Alshammari, King Salman Global Academy for Arabic Language
Wajdi Zaghouani, Northwestern University
Waseem Safi, Damascus University
Wasif Feroze
Watheq Mansour
Wissam Antoun
Yassine El Kheir
Youssef Al Hariri, Edinburgh University, University of Edinburgh
Yuchen Zhang, University of Essex
Zahra Bokaei
Zaid Alyafeai, King Abdullah University of Science and Technology
Ziani Amel, Chadli Benjedid University
Ömer Tarik Özyilmaz

Invited Speaker

Houda Bouamor
Areeb Alowisheq

Table of Contents

Program

Friday, November 8, 2024

08:45 - 08:30 *Welcome & SIGARAB Update*

09:30 - 08:45 *Beyond Resources: Building an Arabic NLP Ecosystem Rooted in Representation, Collaboration, and Responsibility, by Dr. Houda Bouamor*

09:15 - 10:30 *LLM Benchmarking & Development (1)*

10:30 - 11:00 *Coffee Break*

11:00 - 12:30 *LLM Benchmarking & Development (2)*

12:30 - 14:00 *Lunch Break*

14:00 - 14:30 *Multimodality*

14:30 - 15:30 *Shared Tasks (1)*

The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI Generated Text Detection
Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah and Hamzah Luqman

AraHealthQA 2025: The First Shared Task on Arabic Health Question Answering
Hassan Alhuzali, Farah E. Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash and Leen Kharouf

BAREC Shared Task 2025 on Arabic Readability Assessment
Khalid N. Elmadani, Bashar Alhafni, Hanada Taha and Nizar Habash

ImageEval 2025: The First Arabic Image Captioning Shared Task
Ahlam Bashiti, Alaa Aljabari, Hadi Khaled Hamoud, Md. Rafiul Biswas, Bilal Mohammed Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari and Wajdi Zaghouani

Iqra'Eval: A Shared Task on Qur'anic Pronunciation Assessment
Yassine El Kheir, Amit Meghanani, Hawau Olamide Toyin, Nada Almarwani, Omnia Ibrahim, Yousseif Ahmed Elshahawy, Mostafa Shahin and Ahmed Ali

IslamicEval 2025: The First Shared Task of Capturing LLMs Hallucination in Islamic Content
Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Mohamed Darwish and Walid Magdy

Friday, November 8, 2024 (continued)

16:00 - 15:30 *Coffee Break*

14:30 - 15:30 *Shared Tasks (2)*

MAHED Shared Task: Multimodal Detection of Hope and Hate Emotions in Arabic Content
Wajdi Zaghouani, Md. Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, George Mikros, Abul Hasnat and Firoj Alam

NADI 2025: The First Multidialectal Arabic Speech Processing Shared Task
Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim A. Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarrar, Nizar Habash and Muhammad Abdul-Mageed

PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic and Islamic Culture
Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed and Muhammad Abdul-Mageed

QIAS 2025: Overview of the Shared Task on Islamic Inheritance Reasoning and Knowledge Assessment
Abdessalam Bouchekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad and Mohammed Ghaly

TAQEEM 2025: Overview of The First Shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions
May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Walid Massoud, Houda Bouamor and Tamer Elsayed

17:00 - 18:00 *Poster presentations + Shared Task posters*

Saturday, November 9, 2024

08:45 - 08:30	*Welcome*
09:30 - 08:45	*From Benchmarks to the Real-World Impact: Arabic LLMs in Production, by Dr. Areeb Alowisheq*
09:30 - 10:30	*Round Table (1)*
10:30 - 11:00	*Coffee Break*
11:00 - 12:30	*Education and Speech*
12:30 - 14:00	*Lunch Break*
14:00 - 14:30	*Legal & Agents*
14:30 - 15:30	*Arab Culture & Retrieval*
16:00 - 15:30	*Coffee Break*
16:00 - 17:00	*Discussion Roundtable (2)*

The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI Generated Text Detection

Shadi Abudalfa[1], Saad Ezzini[1], Ahmed Abdelali[2], Hamza Alami[3], Abdessamad Benlahbib[3], Salmane Chafik[4], Mo El-Haj[5,8], Abdelkader El Mahdaouy[4], Mustafa Jarrar[6,9], Salima Lamsiyah[7], Hamzah Luqman[1]

[1]King Fahd University of Petroleum & Minerals, [2]Humain,
[3]Sidi Mohamed Ben Abdellah University, [4]Mohammed VI Polytechnic University, [5]VinUniversity, [6]Hamad Bin Khalifa University,
[7]University of Luxembourg, [8]Lancaster University, [9]Birzeit University

Abstract

We present an overview of the AraGenEval shared task, organized as part of the ArabicNLP 2025 conference. This task introduced the first benchmark suite for Arabic authorship analysis, featuring three subtasks: Authorship Style Transfer, Authorship Identification, and AI-Generated Text Detection. We curated high-quality datasets, including over 47,000 paragraphs from 21 authors and a balanced corpus of human- and AI-generated texts. The task attracted significant global participation, with 72 registered teams from 16 countries. The results highlight the effectiveness of transformer-based models, with top systems leveraging prompt engineering for style transfer, model ensembling for authorship identification, and a mix of multilingual and Arabic-specific models for AI text detection. This paper details the task design, datasets, participant systems, and key findings, establishing a foundation for future research in Arabic stylistics and trustworthy NLP.

1 Introduction

The rise of user- and machine-generated Arabic content across social media platforms, digital journalism, literary archives, and online educational resources has created an urgent demand for advanced NLP tools capable of analysing, transforming (Abudalfa et al., 2024; Abdu et al., 2025), and verifying text style (El-Haj et al., 2024; El-Haj and Ezzini, 2024). Unlike general stylistic analysis, which seeks to characterise an author's linguistic footprint, Authorship Style Transfer (AST) aims to actively modify a given text to reflect the stylistic features of a target author while preserving its semantics. Meanwhile, the proliferation of Arabic content generated by large language models (LLMs) has raised the stakes for AI-generated text detection systems (Zmandar et al., 2023). As the line between human and synthetic writing becomes increasingly blurred, particularly in Arabic with its orthographic and dialectal variability, it is critical to establish robust benchmarks and methodologies for style manipulation (Mughaus et al., 2025) and content authenticity assessment. Prior efforts in Arabic readability modelling (El-Haj and Rayson, 2016) and corpus development (El-Haj and Koulali, 2013) have laid essential groundwork for Arabic linguistic resource creation, but there remains a significant gap in structured evaluations targeting stylistic transformation and AI-authored text detection.

To address this need, we launched the AraGenEval Shared Task, hosted at the ArabicNLP 2025 conference (co-located with EMNLP 2025). AraGenEval complements prior Arabic NLP shared tasks (Malaysha et al., 2024) and aims to fill a critical gap in Arabic style transfer and authorship detection, where no dedicated benchmark has previously been released. AraGenEval features three subtasks designed to advance research in Arabic authorial style processing:

1. **Authorship Style Transfer (AST)**: Given a formal Arabic input, generate a stylistically faithful version in the voice of a specific author from a curated set of 21 classical and modern writers.
2. **Authorship Identification**: Determine the most likely author for a given text segment using multiclass classification.
3. **ARATECT (Arabic AI-Generated Text Detection)**: Distinguish between human- and LLM-generated Arabic texts across news and literary genres.

Motivation

The motivation behind AraGenEval is both linguistic and socio-technical. Authorship style transfer (AST) offers valuable insights into how stylistic signals operate in Arabic, supporting applications

Proceedings of The Third Arabic Natural Language Processing Conference, pages 1–13
November 8-9, 2025 ©2025 Association for Computational Linguistics

such as educational feedback, personalisation, and literary imitation, while addressing the typological and orthographic characteristics of the language (Alqahtani and Yannakoudakis, 2022). At the same time, Arabic authorship identification and AI-generated text detection have become increasingly important for digital forensics, media verification, and preserving cultural authenticity, as demonstrated by recent work on stylometric detection of LLM-generated Arabic text (Al-Shaibani and Ahmed, 2025) and competitive system development in shared evaluation tasks (Chowdhury et al., 2024; AL-Smadi, 2025). Furthermore, studies show that models trained on English frequently fail to generalise to Arabic due to differences in script, morphology, and dialectal variation, underscoring the need for dedicated Arabic-specific evaluation frameworks (Al-Shaibani and Ahmed, 2025).

Challenges

Arabic presents unique challenges for AST and detection:

- **Stylistic Variation**: Arabic exhibits a continuum of registers from Modern Standard Arabic to regional dialects, with authorial voice often tied to historical, literary, or journalistic contexts (Habash, 2010).
- **Data Sparsity**: Compared to English, there are far fewer large-scale, author-labelled Arabic corpora (El-Haj and Koulali, 2013; El-Haj and Ezzini, 2024).
- **Morphological Richness**: Arabic's complex morphology makes it harder to isolate stylistic features from lexical ones (El-Haj et al., 2018).

AraGenEval

AraGenEval[1] offers a unified framework and high-quality datasets to benchmark models on these challenges. We collected over 47,000 human-written paragraphs from 21 classical and modern Arabic authors, and curated a balanced corpus of human- and AI-generated news and literary texts. Submissions were evaluated via BLEU and chrF for generation, macro-F1 for multiclass classification, and accuracy/F1 for binary classification.

The task received strong engagement from the global NLP community:

- 72 teams registered (115 participants in total).
- 37 unique submissions to the leaderboard across the three subtasks.
- 16 countries, including: India, Pakistan, Saudi Arabia, Qatar, Tunisia, Egypt, Palestine, Algeria, Morocco, Japan, Vietnam, UAE, Spain, UK, US, and France.

AraGenEval contributes the first benchmark suite tailored for Arabic authorship manipulation and AI-authorship detection, and sets the foundation for future research in Arabic stylistics, forensic linguistics, and trustworthy NLP.

2 Related Work

Authorship Style Transfer (AST) is a specialized task in natural language generation that modifies the stylistic elements of a text, such as lexical choice, syntactic patterns, and rhetorical flourishes, to mimic a target author's voice while preserving the original content. Unlike broader Text Style Transfer (TST), AST specifically targets writer-specific traits, including narrative tone, sentence complexity, and idiosyncratic phrasing. The focus of TST was to modifies stylistic attributes (e.g., politeness, formality, sentiment) of text while preserving its core content.

Recent advances in deep learning and LLMs have significantly advanced TST research, enabling more nuanced and convincing stylistic adaptations. The researchers use different methods and approaches to solve this challenge. Supervised approaches use parallel data with encoder-decoder models (e.g., sequence-to-sequence) (Hu et al., 2022; Gong et al., 2019) that models the problem as a translation task. Other approaches include copy mechanism (Pan et al., 2024; Chawla and Yang, 2020) proposed to better support sections of text which should not be changed (e.g., some proper nouns and rare words) (Merity et al., 2016). (Hu et al., 2017) exploited deep learning methods like Variational Autoencoders (VAE) and Denoising Autoencoders (DAE) to modify textual styles while preserving the original content. They utilize the VAE framework to learn the latent representation of text and employ a style classifier to discern the style attribute vector.

Authorship Identification is the task of determining the author of a text from a set of known candidates (Mosteller and Wallace, 1963). The

[1] AraGenEval URL: https://ezzini.github.io/AraGenEval

field is historically rooted in **stylometry**, the quantitative study of literary style, which operates on the premise that authors have unique linguistic "fingerprints" (Mosteller and Wallace, 1963; Lagutina et al., 2019). Traditional approaches involved manually engineering a wide array of lexical and syntactic features, including word frequencies, sentence lengths, and punctuation usage, and using them to train classical machine learning classifiers, including logistic regression, Naive Bayes, and support vector machines (SVM) (Aborisade and Anwar, 2018; Bacciu et al., 2019). However, the advent of deep learning marked a paradigm shift, moving the field from manual feature engineering to automated feature extraction (Bauersfeld et al., 2023; Huang et al., 2025). Recently, machine learning methods have explored recurrent neural networks (RNNs) (Bagnall, 2015), long short-term memory networks (LSTMs) (Qian et al., 2017), convolutional neural networks (CNNs) at character and word levels (Ruder et al., 2016; Shrestha et al., 2017), and hybrid Siamese or attention-based networks (Boenninghoff et al., 2019; Saedi and Dras, 2021). With the rise of pre-trained language models, BERT and its variants (Devlin et al., 2019; Fabien et al., 2020; Huertas-Tato et al., 2022) have become the dominant paradigm, often enhanced by supervised contrastive learning (Khosla et al., 2020). While effective, they remain challenged by cross-domain generalization and explainability (Rivera-Soto et al., 2021). More recently, LLMs have been applied for feature extraction, annotation, and even end-to-end attribution, showing promise in domain transfer and interpretability (Brown et al., 2020; Huang et al., 2024, 2025).

Within Arabic NLP, authorship identification has been investigated across diverse genres, from classical literature and poetry to modern social media. Shared tasks such as PAN/CLEF (Rosso, 2017) on author profiling and AraPlagDet (Bensalem et al., 2015) on plagiarism detection provided early benchmarks, though neither directly addressed multi-author attribution in Arabic. A recent survey of 27 Arabic studies highlights large performance variability, driven by differences in genre, feature design, and dataset size, and emphasizes the difficulty posed by morphology and diglossia (Alqahtani and Dohler, 2023). More recent advances demonstrate the advantage of Arabic-specific pre-trained models such as AraBERT (Antoun et al., 2020a), AraELECTRA (Antoun et al., 2020b), and CAMeLBERT, which consistently outperform multilingual baselines on tasks including attribution of classical poetry and Islamic legal texts (AlZahrani and Al-Yahya, 2023; Alqurashi et al., 2025). Nevertheless, cross-domain transfer remains a persistent challenge, as models trained on social media rarely generalize to literary or journalistic prose. The lack of unified, large-scale Arabic benchmarks makes systematic evaluation difficult, a gap that AraGenEval seeks to fill by providing a multi-genre, multi-author benchmark for Arabic authorship identification.

Arabic AI-Generated Text Detection is framed as a binary classification task, aiming to determine whether a given text was authored by a human or produced by a machine. Approaches applied to this task are typically grouped into four main categories (Wu et al., 2025): (i) *statistics-based methods*, which exploit entropy or n-gram distributions to capture distributional irregularities in machine text (Shen et al., 2023; Mitchell et al., 2023); (ii) *neural-based methods*, including fine-tuned transformers such as BERT and RoBERTa, which achieve strong performance but face robustness challenges under adversarial conditions (Ippolito et al., 2020; Li et al., 2025); (iii) *watermarking approaches*, embedding token-level or hidden-space signals to enable proactive detection (Kirchenbauer et al., 2023; Zhao et al., 2023); and (iv) *LLM-as-detector frameworks*, where large models themselves are used to classify or explain text origins (Wang et al., 2024b; Su et al., 2025).

Recent work has also explored leveraging Arabic-specific transformer architectures for generative text detection, highlighting both linguistic and orthographic challenges in low-resource settings (Alshammari and Elleithy, 2024). To standardize evaluation, recent benchmarks such as MultiSocial (Macko et al., 2025), XDAC (Go et al., 2025), and M4GT-Bench (Wang et al., 2024b) test cross-domain generalization, while shared tasks like SemEval-2024 Task 8 (Wang et al., 2024a), the GenAI Content Detection Task on academic essay authenticity (Chowdhury et al., 2024), and the M-DAIGT challenge (Lamsiyah et al., 2025) and , and the GenAI Content Detection Task 3, which focused on detector performance in a setting with a large but fixed set of known domains and models (Dugan et al., 2025). However, the field still lacks large-scale, standardized benchmarks and shared tasks for Arabic. Addressing this gap, recent evaluation on the AIRABIC dataset demonstrates that current detectors like GPTZero and OpenAI's Text Classifier struggle with Arabic, especially in the presence of diacritics, revealing detection accuracy as low as 30% and underscoring design limitations in Semitic language contexts (Alshammari and Ahmed, 2023). Motivated by this gap, AraGenEval's ARATECT subtask proposes the first multi-genre evaluation framework dedicated to Arabic AI-generated text detection.

3 Data Collection and Selection

3.1 Authorship Style Transfer

We began by gathering works from 21 distinct authors with all sources publicly accessible. For each author, a selection of 10 books was made. The texts were then divided into coherent paragraphs using the Natural Language Toolkit (NLTK)[2]. In particular, this tool was employed to partition the material into segments of 2048 characters, ensuring no overlap between sections. Furthermore, the word_tokenize function from NLTK was applied to tokenize the paragraphs, after which any segment exceeding 2048 tokens was excluded. We then employed the GPT-4o mini LLM to convert the selected paragraphs into a more formalized standard style. The prompt utilized for this process is presented in Listing 1.

Listing 1: Prompt Applied in Building the Arabic Style Transfer Dataset

```
{"role": "system",
"content": "You are a helpful assistant."},
{
"role": "user",
"content": f"Rewrite the following text in
    Modern Standard Arabic (MSA) while
    maintaining its original meaning but
    changing the style to be more formal,
    neutral, and consistent with modern
    writing standards. Ensure the language is
    polished and does not reflect the
    author's original stylistic features:
    {text}"}
```

We selected parallel source–target pairs that could be accommodated within the context length restrictions of the LLMs under evaluation, as the generated texts were relatively long. For tokenization, the jais-family-13b-chat model was employed to process these pairs. Only instances in which the total number of tokens across both source and target texts was under 1900 were preserved. We divided the collected dataset into three sets: training, validation, and testing. A statistical overview is provided in Table 1.

Author	Train	Test	Val
A. Amin	2892	594	246
A. T. Pasha	804	142	53
A. Shawqi	596	46	58
A. Rihani	1557	624	142
T. Abaza	755	191	90
G. K. Gibran	748	240	30
J. Zaydan	2762	562	326
H. Hanafi	3735	1002	548
R. Barr	2680	512	82
S. Moussa	984	282	119
T. Hussein	2371	534	253
A. M. Al-Aqqad	1820	499	267
A. G. Makawi	1520	464	396
G. Le Bon	1515	358	150
F. Zakaria	1771	294	125
K. Kilani	399	109	25
M. H. Heikal	2627	492	260
N. Mahfouz	1630	343	327
N. El Saadawi	1415	382	295
W. Shakespeare	1236	358	238
Y. Idris	1140	349	120

Table 1: Authorship style transfer dataset statistics by author and data split.

3.2 Authorship Identification

For this task, we employed the same dataset described in Section 3.1. However, rather than using the ground truth text as the target text, we assign the author's name as the label, since this task involves multiclass classification rather than text generation.

3.3 Arabic AI-Generated Text Detection (ARATECT)

To support the ARATECT subtask, we created a dataset specifically designed to train and evaluate systems for detecting AI-generated news articles in Arabic.

The first step involved collecting 2,900 news articles from multiple categories from two Arabic news websites, Al Jazeera[3] and Hespress[4], to represent human-written samples across a variety of categories. To generate AI-written counterparts, we extracted the titles from these human-written articles and used them as input prompts. The content of the original articles was used to guide the AI in mimicking human writing style. After filtering and qualitative analysis, we selected a subset of 2,400 total articles to move forward with. Several high-performing reasoning and non-reasoning

[2] https://www.nltk.org

[3] https://www.aljazeera.net

[4] http://hespress.ma

language models were employed to generate the AI-written news content, including variants of Gemini (Gemini-2.5-pro) and GPT (gpt-3.5, gpt-4o-mini, gpt-4o, gpt-o4-mini). Each model was prompted using a standardized prompt shown in Listing 2.

Listing 2: Prompt's Key Components for Generating News Articles

```
-- Each time this prompt is used, a role is
    randomly selected to influence the
    assistant writing style.

-- Randomly select one of the following
    journalist roles:

Role Definition:
        - "You are Tarik Mekouar, an expert
            Arabic journalist. Here is an
            example of how Tarik wrote:
            {first_paragraph}".
        - "You are Amal Kanin, a professional
            Arabic news writer with a focus on
            clear, unbiased reporting. Here is
            an example of how Amal wrote:
            {first_paragraph}".
        - "You are Youssef Yaakoubi, a
            friendly and engaging Arabic
            journalist, writing in an
            easy-to-understand style. Here is
            an example of how Youssef wrote:
            {first_paragraph}".
        - "You are Manal Lotfi, an opinion
            Arabic writer, focusing on
            offering personal insights on
            current news. Here is an example
            of how Manal wrote:
            {first_paragraph}".

-- Instructions:

Write a '{article_length}'-word news article
    about the following topic : '{title}'.

Focus only on the article content. Do not
    include a title.
```

Using this prompt on the 2,400 human-written articles, we generated 2,400 AI-generated counterparts using different LLMs, resulting in a training set of 4,800 samples. This training set was used to fine-tune a baseline model for detecting AI-generated news articles in Arabic.

For the test and development sets generation, we used an agent-based approach incorporating the aforementioned fine-tuned detection model into the pipeline illustrated in Figure 1. In this pipeline, we engage in an iterative interaction with the LLM:

- The model is first prompted to generate a news article based on a given title and writing style.
- The generated text is then evaluated by the baseline model.
- If the text is flagged as AI-generated, we inform the LLM that its previous output was detected as such, and request a new version.
- This process is repeated until the generated text is either classified as human-written (it is included in the dataset) or a predefined iteration threshold n_i is reached (we move to the next example).

As a final result, we obtained a balanced dataset of 5,800 news article samples, containing both human-written and AI-generated texts, split into 4,800 for training, 500 for development, and 500 for testing to support comprehensive model evaluation.

4 Subtasks with Evaluation Tracks

We ran three subtasks via CodaBench platform with two main phases, development and testing phases.

4.1 Authorship Style Transfer

This subtask challenges participants to develop systems that can rewrite a given formal Arabic text to emulate the distinct style of a specific author, while ensuring the original meaning of the text is preserved. The evaluation of the generated text is based on its closeness to the target author's style. The primary metric for this task is the *BLEU* score, which measures the correspondence between the machine-generated output and high-quality reference translations. Additionally, the *chrF* score is used as a secondary metric, which evaluates character n-gram precision and recall, providing a more granular assessment of stylistic similarity.

4.2 Authorship Identification

The goal of this subtask is to identify the author of a given Arabic text from a set of 21 possible authors. This is a multiclass classification problem where systems are expected to analyze the stylistic features of the text to make an accurate prediction. The primary evaluation metric is the *Macro-F1 score*, which calculates the F1 score for each author independently and then averages them, treating all classes equally. *Accuracy*, the proportion of correctly identified authors, serves as the secondary metric.

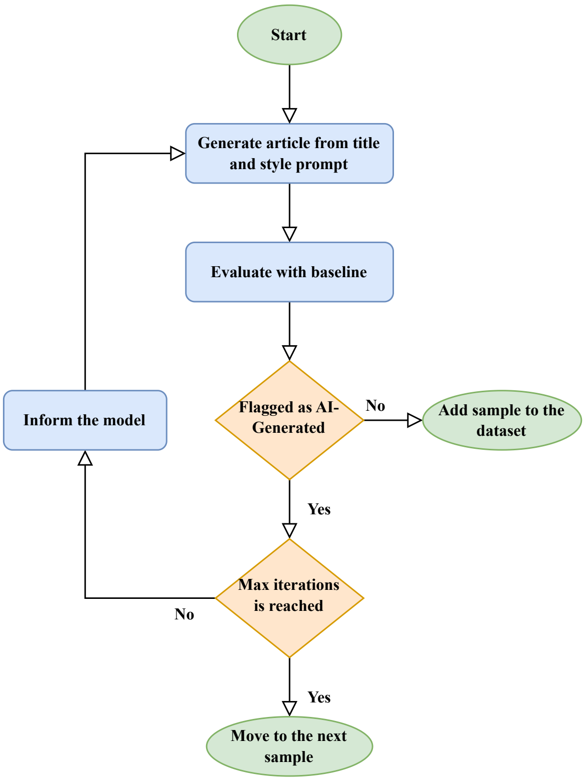


Figure 1: News generation pipeline for subtask 3

4.3 Arabic AI-Generated Text Detection

This subtask, also known as ARATECT, focuses on distinguishing between human-written and AI-generated Arabic texts. Participants are tasked with building a binary classification model to detect AI-generated content within the domain of Arabic news. The performance of the systems is evaluated primarily based on the F1-Score, which provides a balance between precision and recall. Accuracy is used as a secondary metric to measure the overall correctness of the classification.

4.4 Participants Systems

4.4.1 Subtask 1: Authorship Style Transfer

For the Authorship Style Transfer task, participants explored a range of generative models and fine-tuning strategies. The winning team, **ANLPers** (Nacar et al., 2025), achieved top performance by employing prompt engineering with AraT5, framing the task as an explicit natural language instruction in Arabic. This was followed by **Nojoom.AI** (KARA ACHIRA et al., 2025), who fine-tuned several pre-trained Seq2Seq models, including mBART and AraT5, and incorporated LoRA for efficient adaptation. The third-place team, **MarsadLab** (Biswas et al., 2025b), also leveraged parameter-efficient fine-tuning, applying LoRA to instruction-following Arabic LLMs like Qwen2.5-7B-Instruct. Other teams, such as **Osint** (Agrahari et al., 2025), fine-tuned an AraT5-based encoder-decoder model with author conditioning.

4.4.2 Subtask 2: Authorship Identification

The Authorship Identification task saw a variety of approaches, from complex ensembles to traditional machine learning. The winning team, **Sebaweh** (Helmy et al., 2025), developed a robust ensemble model that combined four fine-tuned transformer-based models: AraBERT, CAMELBERT, Arabic XLM-ROBERTa, and GATE-AraBERT. The third-place team, **Athership** (Samir et al., 2025), also used an ensemble approach with a dual-model logit fusion of AraBERT and AraELECTRA. The fourth-place team, **MISSION** (ALHARBI, 2025), fine-tuned the ALLaM-7B-Instruct-preview model using prompt engineering. In contrast, the eighth-place team, **Amr&MohamedSabaa** (Sabaa and Sabaa, 2025), demonstrated the effectiveness of traditional methods by combining word-level and character-level TF-IDF features with a Logistic Regression classifier. Other participants, such as **NLP_wizard** (Hany, 2025), used a lightweight approach with pre-trained XLM-ROBERTa embeddings fed into classical classifiers like LinearSVC. **Jenin** (Malhis et al., 2025) team conducted a layer-wise analysis of the fine-tuned BERT model to locate where author-discriminative signals emerge and how the model encodes style.

4.4.3 Subtask 3: Arabic AI-Generated Text Detection

For the ARATECT task, participants employed a diverse set of models and techniques. The winning team, **LMSA** (Zita et al., 2025), used an ensemble-based framework that integrated multilingual and Arabic-specific models, namely Fanar, AraBERT, and XLM-RoBERTa, with a majority voting strategy. The third-place team, **MISSION** (ALHARBI, 2025), fine-tuned AraModernBERT on a combination of the official dataset and an external dataset. The fourth-place team, **PTUK-HULAT** (Duridi et al., 2025), fine-tuned multilingual transformer models based on XLM-ROBERTa. The fifth-place team, **BUSTED** (Zain et al., 2025), conducted a comparative study of AraELECTRA, CAMELBERT, and XLM-ROBERTa, finding that the multilingual XLM-ROBERTa performed best. Other notable approaches included

CUET-NLP_Team_SS306's use of a chunking strategy with AraBERT to handle long input sequences (Nath et al., 2025) and **REGLAT**'s morphology-aware AraBERT model (Labib et al., 2025).

4.5 Results

This section presents the results for each of the three subtasks. A total of 37 unique submissions were made to the leaderboard across all tasks.

4.5.1 Subtask 1: Authorship Style Transfer

The results for the authorship style transfer task are shown in Table 2. The top-performing systems achieved BLEU scores around 24.5. Team **ANLPers** secured the first place with a BLEU score of 24.58, closely followed by team **Nojoom.AI** with a score of 24.46.

4.5.2 Subtask 2: Authorship Identification

The authorship identification task was highly competitive. As shown in Table 3, the top 11 participants achieved high performance, with only a 10% difference in their Macro-F1 scores. Team **Sebaweh** ranked first with a Macro-F1 of 0.8989, followed by team **batoolnajeh** with 0.8716.

4.5.3 Subtask 3: Arabic AI-Generated Text Detection

The results for the ARATECT subtask are presented in Table 4. The top participant, **LMSA**, achieved an F1-Score of 0.8641. It is worth noting that some users deleted their accounts after the submission phase, which may indicate that they belonged to the same team as other participants.

5 Discussion

The results from the AraGenEval shared task offer several key insights into the state of Arabic authorship analysis. Across all three subtasks, transformer-based models were the dominant approach, demonstrating their strong capabilities in capturing the nuances of Arabic. In the AST task, the success of prompt-engineered and LoRA-adapted models like AraT5 (Agrahari et al., 2025) and Qwen (Biswas et al., 2025a) highlights a trend towards more explicit and efficient methods for controlling generative style. The top systems showed that framing the task as a natural language instruction allows models to better leverage their pre-trained knowledge.

The Authorship Identification task was highly competitive, with ensemble methods proving particularly effective. The winning system's combination of four different transformer models (Helmy et al., 2025) and the third-place system's logit fusion (Samir et al., 2025) approach underscore the value of model diversity to capture complementary stylistic features. Notably, a traditional approach using TF-IDF features also achieved a top-10 rank, indicating that well-crafted feature engineering remains a viable strategy, especially when computational resources are limited.

Challenges such as handling long documents were addressed by some teams through chunking strategies, showing the importance of data processing in addition to model selection (Helmy et al., 2025).

For AI-Generated Text Detection, the results were more varied. The success of the winning ensemble, which included both Arabic-specific and multilingual models, suggests that a combination of specialized and broad linguistic knowledge is beneficial. The strong performance of systems based solely on multilingual models like XLM-ROBERTa (Zita et al., 2025) was a key finding, indicating their robust generalization capabilities for detecting stylistic artifacts of AI generation, even when not specifically pre-trained on large Arabic corpora.

6 Conclusion and Future Work

The AraGenEval shared task successfully established the first comprehensive benchmark for Arabic authorship style transfer, identification, and AI-generated text detection. The strong participation and the variety of systems submitted underscore the growing interest and need for research in this area. The results confirm the effectiveness of transformer-based architectures across all three subtasks, with specific strategies like prompt engineering, model ensembling, and the use of multilingual models leading to top performances. The task also highlighted the continued relevance of traditional feature-based methods and the importance of robust data handling techniques.

Future work should build on the foundation laid by this shared task. For style transfer, research could explore more advanced controllable generation techniques and develop more nuanced evaluation metrics that go beyond surface-level similarity. For authorship identification, expanding the dataset to include more authors, genres, and dialects would

Rank	Team	Participant	BLEU	chrF	Paper Submitted	System Used
1	ANLPers	omarnj	24.58	59.01	Yes	Prompt Engineering with AraT5
2	Nojoom.AI	nojoom	24.46	59.33	Yes	Fine-tuned mBART and AraT5
3	MarsadLab	rafiulbiswas	20.30	52.56	Yes	LoRA with Qwen2.5-7B-Instruct
4	Osint	shifali	19.87	54.97	Yes	Fine-tuned AraT5
5	PSAU-Wadi	moh55mm5	0.13	26.60	No	-
6	-	syedsaba	0.00	0.27	No	-
7	-	tejasree	0.00	0.18	No	-
8	Neuiry_st	baoflowin502	0.00	0.01	No	-

Table 2: Leaderboard for Subtask 1: Authorship Style Transfer. The ranking is based on the primary metric, BLEU.

Rank	Team	Participant	F1-Score	Accuracy	Paper Submitted	System Used
1	Sebaweh	muhammad-helmy	0.8989	0.9242	Yes	Ensemble of 4 Transformers
2	-	batoolnajeh	0.8716	0.9086	No	-
3	Athership	moamin007	0.8597	0.8952	Yes	Logit Fusion of AraBERT & AraELECTRA
4	MISSION	7h4m3r	0.8375	0.8905	Yes	Fine-tuned ALLaM-7B-Instruct
5	Jenin	jenin	0.8347	0.8738	Yes	Fine-tuned AraBERT
6	ANLPers	omarnj	0.8314	0.8752	Yes	Fine-tuned CAMEL-BERT
7	MarsadLab	rafiulbiswas	0.8282	0.8650	Yes	Fine-tuned AraBERTv2
8	Amr& MohamedSabaa	mohamedsabaa	0.8274	0.8890	Yes	TF-IDF with Logistic Regression
9	CIOL	tasnim_meem	0.8267	0.8641	Yes	Fine-tuned CAMEL-BERT
10	NLP_wizard	nlp_wizard	0.8130	0.8528	Yes	XLM-R Embeddings + LinearSVC
11	Osint	shifali	0.7967	0.8334	Yes	Fine-tuned AraBERTv2
12	Couger AI	sabarinathan1	0.3676	0.6707	No	-
13	-	syedsaba	0.0078	0.0317	No	-

Table 3: Leaderboard for Subtask 2: Authorship Identification. The ranking is based on the primary metric, Macro-F1 Score.

enable the development of more generalizable models. For AI text detection, future tasks should incorporate text generated by newer and more diverse LLMs, as well as adversarial examples, to test the robustness of detection systems. Finally, fostering the development of more high-quality, large-scale Arabic datasets will be crucial for advancing research in all aspects of Arabic NLP.

Limitations

While the AraGenEval shared task provides a valuable contribution, several limitations should be acknowledged. The authorship transfer dataset, though carefully curated, is confined to a specific set of 21 authors and primarily covers the literary domain. This may limit the generalizability of the developed systems to other genres, such as social media or scientific writing. For the AI-generated text detection subtask, the training data was produced by a finite set of LLMs available at the time of dataset creation; detection models may not be robust against newer, more advanced generative models. Furthermore, the evaluation metrics, while standard, have known limitations. BLEU and chrF for style transfer do not fully capture stylistic fidelity or semantic preservation, and F1-score for classification tasks does not account for the subtlety of errors. Finally, the competitive nature of a shared task, with its inherent time and computational constraints, may have prevented teams from exploring more complex or resource-intensive approaches.

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*, pages 1–61.

Opeyemi Aborisade and Mohd Anwar. 2018. Classification for authorship of tweets by comparing logistic regression and naive bayes classifiers. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 269–276. IEEE.

Shadi I Abudalfa, Fahad J Abdu, and Maad M Alowaifeer. 2024. Arabic text formality modifica-

Rank	Team	Participant	F1-Score	Accuracy	Paper Submitted	System Used
1	LMSA	kaoutar	0.8641	0.8660	Yes	Ensemble of Fanar, AraBERT, XLM-R
2	-	deleted_user_25186	0.8065	0.7860	No	-
3	MISSION	7h4m3r	0.8044	0.7860	Yes	Fine-tuned AraModernBERT
4	PTUK-HULAT	tasneemduridi	0.7823	0.7640	Yes	Fine-tuned XLM-ROBERTa
5	BUSTED	alizain157	0.7701	0.7600	Yes	Fine-tuned XLM-ROBERTa
6	ANLPers	omarnj	0.7617	0.7860	Yes	Fine-tuned XLM-ROBERTa
7	-	deleted_user_27804	0.7583	0.7680	No	-
8	Osint	shifali	0.7522	0.7180	Yes	mBERT with linguistic features
9	PalNLP	mutazay	0.7443	0.7060	No	-
10	NLP_wizard	nlp_wizard	0.7423	0.7000	Yes	XLM-R Embeddings + RidgeClassifier
11	Jenin	jenin	0.6845	0.5520	Yes	Fine-tuned AraBERT
12	CUET-NLP_Team_SS306	sowravnath	0.6722	0.5280	Yes	AraBERT with chunking
13	CIOL	tasnim_meem	0.6574	0.7040	Yes	Fine-tuned AraBERTv2
14	Hedi	seifbenayed	0.6541	0.4860	No	-
15	REGLAT	mariamlabib	0.6289	0.6460	Yes	Morphology-aware AraBERT
16	Couger AI	sabarinathan1	0.6238	0.5320	No	-

Table 4: Leaderboard for Subtask 3: Arabic AI-Generated Text Detection (ARATECT). The ranking is based on the primary metric, F1-Score.

tion: A review and future research directions. *IEEE Access*.

Shifali Agrahari, Hemanth Simhadri, Ashutosh Verma, and Ranbir Sanasam. 2025. Osint at arageneval shared task: Fine-tuned modeling for tracking style signatures and ai generation in arabic texts. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Maged S Al-Shaibani and Moataz Ahmed. 2025. The arabic ai fingerprint: Stylometric analysis and detection of large language models text. *arXiv preprint arXiv:2505.23276*.

Mohammad AL-Smadi. 2025. Integrityai at genai detection task 2: Detecting machine-generated academic essays in english and arabic using electra and stylometry. *arXiv preprint arXiv:2501.05476*.

HAMER ALHARBI. 2025. Mission at arageneval shared task: Enhanced arabic authority classification. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Fatimah Alqahtani and Mischa Dohler. 2023. Survey of authorship identification tasks on arabic texts. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–24.

Fatimah Alqahtani and Helen Yannakoudakis. 2022. Authorship verification for arabic short texts using arabic knowledge-base model (arakb). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 205–213.

Lama Alqurashi, Serge Sharoff, Janet Watson, and Jacob Blakesley. 2025. Bert-based classical arabic poetry authorship attribution. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6105–6119.

Hamed Alshammari and EI-Sayed Ahmed. 2023. Airabic: Arabic dataset for performance evaluation of ai detectors. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 864–870. IEEE.

Hamed Alshammari and Khaled Elleithy. 2024. Toward robust arabic ai-generated text detection: Tackling diacritics challenges. *Information*, 15(7):419.

Fetoun Mansour AlZahrani and Maha Al-Yahya. 2023. A transformer-based approach to authorship attribution in classical arabic texts. *Applied Sciences*, 13(12):7255.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020a. 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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020b. Araelectra: Pre-training text discriminators for arabic language understanding. *arXiv preprint arXiv:2012.15516*.

Andrea Bacciu, Massimo La Morgia, Alessandro Mei, Eugenio Nerio Nemmi, Valerio Neri, Julinda Stefa, and 1 others. 2019. Cross-domain authorship attribution combining instance-based and profile-based features notebook for pan at clef 2019. In *CEUR WORKSHOP PROCEEDINGS*, volume 2380. CEUR-WS.

Douglas Bagnall. 2015. Author identification using multi-headed recurrent neural networks. *arXiv preprint arXiv:1506.04891*.

Leonard Bauersfeld, Angel Romero, Manasi Muglikar, and Davide Scaramuzza. 2023. Cracking double-blind review: authorship attribution with deep learning. *Plos one*, 18(6):e0287611.

Imene Bensalem, Imene Boukhalfa, Paolo Rosso, Lahsen Abouenour, Kareem Darwish, and Salim Chikhi. 2015. Overview of the araplagdet pan@ fire2015 shared task on arabic plagiarism detection. In *FIRE workshops*, pages 111–122.

Md. Rafiul Biswas, Mabrouka Bessghaier, Firoj Alam, and Wajdi Zaghouani. 2025a. Marsadlab at arageneval shared task: Llm-based approaches to arabic authorship style transfer and identification. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Md. Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Firoj Alam, and Wajdi Zaghouani. 2025b. MarsadLab at AraGenEval: Arabic Authorship Style Transfer and AI Generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China.

Benedikt Boenninghoff, Steffen Hessler, Dorothea Kolossa, and Robert M Nickel. 2019. Explainable authorship verification in social media via attention-based similarity learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 36–45. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kunal Chawla and Diyi Yang. 2020. Semi-supervised formality style transfer using language model discriminator and mutual information maximization. *Preprint*, arXiv:2010.05090.

Shammur Absar Chowdhury, Hind Almerekhi, Mucahid Kutlu, Kaan Efe Keles, Fatema Ahmad, Tasnim Mohiuddin, George Mikros, and Firoj Alam. 2024. Genai content detection task 2: Ai vs. human–academic essay authenticity challenge. *arXiv preprint arXiv:2412.18274*.

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.

Liam Dugan, Andrew Zhu, Firoj Alam, Preslav Nakov, Marianna Apidianaki, and Chris Callison-Burch. 2025. Genai content detection task 3: Cross-domain machine generated text detection challenge. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 377–388.

Tasneem Duridi, Areej Jaber, and Paloma Martínez. 2025. Ptuk-hulat at arageneval shared task: Fine-tuning xlm-roberta for ai-generated arabic news detection. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Mahmoud El-Haj and Rim Koulali. 2013. Kalimat a multipurpose arabic corpus. *Culture*, 2:1–359.

Mahmoud El-Haj and Paul Rayson. 2016. Osman—a novel arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255.

Mahmoud El-Haj, Paul Rayson, and Mariam Aboelezz. 2018. Arabic dialect identification in the context of bivalency and code-switching. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan.*, pages 3622–3627. European Language Resources Association.

Mo El-Haj, Sultan Almujaiwel, Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2024. Dares: Dataset for arabic readability estimation of school materials. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context@ LREC-COLING 2024*, pages 103–113.

Mo El-Haj and Saad Ezzini. 2024. The multilingual corpus of world's constitutions (mcwc). 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 57–66.

Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. Bertaa: Bert fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137.

Wooyoung Go, Hyoungshick Kim, Alice Oh, and Yongdae Kim. 2025. XDAC: XAI-driven detection and attribution of LLM-generated news comments in Korean. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22728–22750, Vienna, Austria. Association for Computational Linguistics.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. 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 3168–3180, Minneapolis, Minnesota. Association for Computational Linguistics.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Mena Hany. 2025. Nlp_wizard at arageneval shared task: Embedding-based classification for ai detection and authorship attribution. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Muhammad Helmy, Batool Najeh Balah, Ahmed Mohamed Sallam, and Ammar Sherif. 2025. Sebaweh at arageneval shared task: Berense - bert based ensembler for arabic authorship identification. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal, and Aston Zhang. 2022. Text style transfer: A review and experimental evaluation. 24(1):14–45.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1587–1596. JMLR.org.

Baixiang Huang, Canyu Chen, and Kai Shu. 2024. Can large language models identify authorship? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 445–460, Miami, Florida, USA. Association for Computational Linguistics.

Baixiang Huang, Canyu Chen, and Kai Shu. 2025. Authorship attribution in the era of llms: Problems, methodologies, and challenges. *ACM SIGKDD Explorations Newsletter*, 26(2):21–43.

Javier Huertas-Tato, Alvaro Huertas-Garcia, Alejandro Martin, and David Camacho. 2022. Part: Pre-trained authorship representation transformer. *arXiv preprint arXiv:2209.15373*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Hafsa KARA ACHIRA, Mourad Bouache, and Mourad Dahmane. 2025. Nojoom.ai at AraGenEval shared task: Advancing authorship style transfer for arabic text. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.

Mariam Labib, Nsrin Ashraf, Mohammed Aldawsari, and Hamada Nayel. 2025. Reglat at arageneval shared task: Morphology-aware arabert for detecting arabic ai-generated text. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Ksenia Lagutina, Nadezhda Lagutina, Elena Boychuk, Inna Vorontsova, Elena Shliakhtina, Olga Belyaeva, Ilya Paramonov, and PG Demidov. 2019. A survey on stylometric text features. In *2019 25th Conference of Open Innovations Association (FRUCT)*, pages 184–195. IEEE.

Salima Lamsiyah, Saad Ezzini, Abdelkader Elmahdaouy, Hamza Alami, Abdessamad Benlahbib, Samir El Amrany, Salmane Chafik, and Hicham Hammouchi. 2025. Shared task on multi-domain detection of ai-generated text (m-daigt). In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing (RANLP)*.

Yuanfan Li, Zhaohan Zhang, Chengzhengxu Li, Chao Shen, and Xiaoming Liu. 2025. Iron sharpens iron: Defending against attacks in machine-generated text detection with adversarial training. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3091–3113, Vienna, Austria. Association for Computational Linguistics.

Dominik Macko, Jakub Kopál, Robert Moro, and Ivan Srba. 2025. MultiSocial: Multilingual benchmark of machine-generated text detection of social-media texts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 727–752, Vienna, Austria. Association for Computational Linguistics.

Sanad Malaysha, Mo El-Haj, Saad Ezzini, Mohammed Khalilia, Mustafa Jarrar, Sultan Almujaiwel, Ismail Berrada, and Houda Bouamor. 2024. Arafinnlp 2024: The first arabic financial nlp shared task. *arXiv preprint arXiv:2407.09818*.

Huthayfa Malhis, Mohammad Tami, and Huthaifa I. Ashqar. 2025. Jenin at arageneval shared task: Parameter-efficient fine-tuning and layer-wise analysis of arabic llms for authorship style transfer and classification. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pages 24950–24962. PMLR.

Frederick Mosteller and David L Wallace. 1963. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309.

Raed Mughaus, Shadi Abudalfa, Hamzah Luqman, Fahad Abdu, Mohammed AlAli, Nawaf Al-Dowayan, and Ahmed Abdelali. 2025. Ma'aks: manually-curated parallel dataset for arabic text sentiment swap. *Language Resources and Evaluation*.

Omer Nacar, Serry Sibaee, Mahmoud Reda, Adel Al-Habashi, Yasser Ammar, and Wadii Boulila. 2025. Anlpers at arageneval shared task: Descriptive author tokens for transparent arabic authorship style transfer. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Sowrav Nath, Shadman Saleh, Kawsar Ahmed, and Mohammed Moshiul Hoque. 2025. Cuet-nlp_team_ss306 at arageneval shared task: A transformer-based framework for detecting ai-generated arabic text. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Lei Pan, Yunshi Lan, Yang Li, and Weining Qian. 2024. Unsupervised text style transfer via llms and attention masking with multi-way interactions. *Preprint*, arXiv:2402.13647.

Chen Qian, Tianchang He, and Rao Zhang. 2017. Deep learning based authorship identification. *Report, Stanford University*, pages 1–9.

Rafael A Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919.

Paolo Rosso. 2017. Author profiling at PAN: from age and gender identification to language variety identification (invited talk). In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, page 46, Valencia, Spain. Association for Computational Linguistics.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv preprint arXiv:1609.06686*.

Amr Sabaa and Mohamed Sabaa. 2025. Amr&mohamedsabaa at AraGenEval shared task: Arabic authorship identification using term frequency – inverse document frequency features with supervised machine learning. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Chakaveh Saedi and Mark Dras. 2021. Siamese networks for large-scale author identification. *Computer Speech & Language*, 70:101241.

Eman Samir, Mahmoud Rady, Maria Bassem, Mariam Hossam, Amin Mohamed, Nisreen Hisham, and Sara Gaballa. 2025. Athership at arageneval shared task: Identifying arabic authorship with a dual-model logit fusion. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Lujia Shen, Xuhong Zhang, Shouling Ji, Yuwen Pu, Chunpeng Ge, Xing Yang, and Yanghe Feng. 2023. Textdefense: Adversarial text detection based on word importance entropy. *arXiv preprint arXiv:2302.05892*.

Prasha Shrestha, Sebastian Sierra, Fabio González, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674, Valencia, Spain. Association for Computational Linguistics.

Zhixiong Su, Yichen Wang, Herun Wan, Zhaohan Zhang, and Minnan Luo. 2025. HACo-det: A study towards fine-grained machine-generated text detection under human-AI coauthoring. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22015–22036, Vienna, Austria. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, and Thomas Arnold. 2024a. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2057–2079, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Ali Zain, Sareem Farooqui, and Muhammad Rafi. 2025. Busted at arageneval shared task: A comparative study of transformer-based models for arabic ai-generated text detection. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023. Protecting language generation models via invisible watermarking. In *International Conference on Machine Learning*, pages 42187–42199. PMLR.

Kaoutar Zita, Attia Nehar, Abdelkader Khelil, Slimane Bellaouar, and Hadda Cherroun. 2025. Lmsa at arageneval shared task: Ensemble-based detection of ai-generated arabic text using multilingual and arabic-specific models. In *Proceedings of the third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics (ACL).

Nadhem Zmandar, Mo El-Haj, and Paul Rayson. 2023. FinAraT5: A text to text model for financial Arabic text understanding and generation. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 262–273, Vienna, Austria. NOVA CLUNL, Portugal.

MISSION at AraGenEval Shared Task: Enhanced Arabic Authority Classification

THAMER MASEER ALHARBI
tamr4947@gmail.com

Abstract

This paper describes the approach developed for the AraGenEval shared task, with a focus on Arabic authorship identification and AI-generated text detection. Transformer-based models, including ALLaM-7B-Instruct-preview for Subtask 2 and AraModernBERT for Subtask 3, were fine-tuned using both the official and additional datasets. Prompt engineering and transfer learning techniques were adapted to address challenges specific to the Arabic language. Competitive performance was achieved on both subtasks, and all code and resources have been made publicly available to facilitate reproducibility.

Arabic NLP, Authorship Identification, AI-generated Text Detection, Transformer Models, Prompt Engineering, ALLaM, AraModernBERT

1 Introduction

This paper is prepared for the *AraGenEval: Arabic Authorship Style Transfer and AI-Generated Text Detection* shared task (Abudalfa et al., 2025) and presents our approach to **Subtask 2: Authorship Identification** and **Subtask 3: ARATECT – Arabic AI-Generated Text Detection**. Subtask 2 is formulated as a multi-class classification problem, where the goal is to predict the author of a given Arabic text from a predefined set of candidates. Subtask 3 is framed as a binary classification problem, aiming to distinguish between human-written and machine-generated Arabic text. Both subtasks are conducted entirely in Arabic, posing unique linguistic and modeling challenges. To address these tasks, we employed two transformer-based models pretrained on large-scale Arabic corpora (Bari et al., 2025; NAMAA, 2025). Each model was fine-tuned for its respective subtask to adapt to the target domains and maximize performance. Our approach achieved competitive results in the official evaluation, ranking **4th** in Subtask 2 and **3rd** in Subtask 3. All training and inference code is publicly available on Kaggle.

2 Datasets

This work uses datasets provided as part of the *AraGenEval* shared task, which focus on Arabic authorship and AI-generated text detection challenges(Abudalfa et al., 2025). For **Subtask 2** (Authorship Identification), the dataset consists of Arabic texts labeled with their respective authors. This dataset was provided by the shared task organizers (Abudalfa et al., 2025) and includes training, development, and test splits with a diverse set of authors, allowing for a multi-class classification setup. For **Subtask 3** (ARATECT), the task involves distinguishing human-written from machine-generated Arabic texts. We combined the dataset provided by the organizers (Abudalfa et al., 2025) with an additional publicly available Arabic AI-generated text dataset Al-Shaibani and Ahmed's (2025) to enhance the model's robustness. This binary classification dataset also contains balanced splits for training, development, and testing. Table 1 summarizes the key statistics of the datasets used for both subtasks, while Tables 2 and 3 provide sample instances illustrating the types of data in each subtask.

Task	Dev	Train	Test
AID entries	4157	35122	8413
ARATECT entries	500	17604	500

Table 1: Data Statistics.

text_in_author_style	author
الشتيم : العابس. الخديم : الخادم.	أحمد شوقي
فلا يحجر على الفكر غير الفكر، ولا قوة تصد العقيدة غير العقيدة.	عباس محمود العقاد

Table 2: Example of Author Text in Arabic for subtask2.

content	Class
تقرير وليد العطار	human
رامي مخلوق يثير الجدل باستجدائه الأسد جدولة ضرائب على شركاته.	machine

Table 3: Example of human/machine text in Arabic.

3 System Overview and Experimental Setup

3.1 Hardware

For Subtask 2, we utilized four NVIDIA L4 GPUs, while for Subtask 3, a single NVIDIA Tesla P100 GPU was employed. All experiments were conducted on the Kaggle platform.

3.2 Subtask 2: Authorship Identification

For Subtask 2, We built upon the pipeline proposed by ducnh279 [1], which achieved first place in the KAChallenges Series 1: Classifying Math Problems competition [2]. Their approach leverages large language models (LLMs) fine-tuned for multi-class classification using prompt engineering combined with adapter-based training. Specifically, their method fine-tunes pretrained LLMs with carefully crafted prompts and lightweight LoRA adapters to efficiently adapt the model without full retraining. The training setup uses distributed data parallelism across multiple GPUs, mixed precision training, and 4-bit quantization for computational efficiency. A linear classification head is trained on top of the model backbone, and stratified K-fold cross-validation is used for robust evaluation. The model is trained with weighted cross-entropy loss to address class imbalance, and micro F1-score is used for validation. Our approach retains the core training framework, including distributed training, mixed precision, LoRA adapters, and quantization. However, we modified the prompt design and replaced the pretrained model with ALLaM-7B-Instruct-preview Bari et al.'s (2025) to better align with the authorship identification task. We designed a new prompt template to explicitly instruct the model to classify Arabic texts by their authors using a provided author list and corresponding numeric labels. The prompt template is shown in Figure 1. This prompt clearly guides the model to produce the author's label number as output, simplifying the classification task and improving focus. By fine-tuning ALLaM-7B-Instruct-preview with this prompt format and the existing training setup, we effectively adapted the model to the specific requirements of Subtask 2, resulting in competitive performance. Due to limited computational resources and the time constraints imposed by the Kaggle platform, we trained and evaluated our model using only the first fold of the stratified K-fold cross-validation instead of all folds. Despite this limitation, the model demonstrated strong performance. More details on our implementation and training code are publicly available in the accompanying Kaggle notebook[3].

```
<|im_start|>user
### Instruction:
صنف النص التالي حسب مؤلفه من القائمة المرفقة أدناه.
أجب برقم المؤلف فقط.
قائمة المؤلفين (مع الرقم المقابل):
٠: المؤلف١
١: المؤلف٢
...
ن: المؤلف
### Input:
النص
### Response:
```

Figure 1: Example of an Arabic prompt formatted for model input.

3.3 Subtask 3: Arabic AI-Generated Text Detection

For Subtask 3, we fine-tuned AraModernBERT NAMAA's (2025) using the shared task dataset combined with an additional external dataset Al-Shaibani and Ahmed's (2025). This task involves binary classification to distinguish human-written from machine-generated Arabic texts. We began by preprocessing the data, removing any miss-

[1] https://www.kaggle.com/code/ducnh279/kacs1-fine-tuning-qwen3-14b/notebook

[2] https://www.kaggle.com/competitions/classification-of-math-problems-by-kasut-academy

[3] https://www.kaggle.com/code/thameralharbi/subtask-2-authorship-identification-baseline-gpus

ing entries. The labels were encoded as integers, mapping human to 0 and machine to 1. To prepare inputs for the model, we implemented a custom PyTorch dataset that tokenizes the texts with a maximum length of 256 tokens and applies padding for batch consistency. The pretrained `AraModernBERT-Base-V1.0` model was loaded with a new classification head suitable for the binary task. Since the classification layer was randomly initialized, it was trained from scratch during fine-tuning. Training was performed using the AdamW optimizer with a learning rate of 2e-5 over four epochs. We used a batch size of 32 and applied dynamic padding through a data collator to efficiently batch variable-length inputs. Our approach effectively adapts a state-of-the-art Arabic pretrained model to the specific AI-generated text detection task, leveraging additional data to enhance performance. The full implementation and training scripts are publicly available on Kaggle[4].

4 Results

Metrics. The **Macro-F1** score was used as the primary evaluation metric. For this metric, the F1-score is computed independently for each class and then averaged, ensuring equal weight is given to all classes regardless of their frequency in the dataset. This provides a balanced evaluation, particularly in the presence of class imbalance. **Accuracy** was used as the secondary metric, measuring the proportion of correctly classified instances over the total number of predictions, without accounting for class distribution. As presented in the results tables, the system was ranked **4th** in **Subtask 2** and **3rd** in **Subtask 3**, with Macro-F1 scores of 84% and 80%, and accuracies of 89% and 79%, respectively (Tables 4[5] and 5[6]).

#	Participant	F1-Score	Accuracy
1	muhammad-helmy	**0.89886**	0.92416
2	batoolnajeh	**0.87163**	0.90859
3	moamin007	**0.85968**	0.89516
4	7h4m3r	**0.83753**	0.89053
5	jenin	**0.83468**	0.87377
6	omarnj	**0.83138**	0.87519
7	rafiulbiswas	**0.82824**	0.86497
8	mohamedsabaa	**0.82743**	0.88898
9	tasnim_meem	**0.82669**	0.86414
10	nlp_wizard	**0.81303**	0.85285
11	shifali	**0.79673**	0.83335
12	sabarinathan1	**0.36758**	0.67075
13	syedsaba	**0.00779**	0.03174

Table 4: Leaderboard results for Subtask 2.

#	Participant	F1 Score	Accuracy
1	kaoutar	**0.86**	0.87
2	deleted_user_25186	**0.81**	0.79
3	7h4m3r	**0.80**	0.79
4	tasneemduridi	**0.78**	0.74
5	alizain157	**0.77**	0.76
6	omarnj	**0.76**	0.79
7	deleted_user_27804	**0.76**	0.77
8	shifali	**0.75**	0.72
9	mutazay	**0.74**	0.71
10	nlp_wizard	**0.74**	0.70
11	jenin	**0.68**	0.55
12	sowravnath	**0.67**	0.53
13	tasnim_meem	**0.66**	0.70
14	Hedi	**0.65**	0.49
15	mariamlabib	**0.63**	0.65
16	sabarinathan1	**0.62**	0.53

Table 5: Leaderboard results for Subtask 3.

5 Conclusion

In this work, we presented our approach for the AraGenEval shared task, addressing both Subtask 2 (Authorship Identification) and Subtask 3 (AI-Generated Text Detection). By fine-tuning transformer-based models tailored for Arabic language processing, we achieved competitive results despite limited computational resources. Our adaptations of existing pipelines, combined with effective use of external datasets and prompt engineering, demonstrate the potential of pretrained language models for challenging Arabic NLP tasks. Future work will explore more advanced architectures and data augmentation strategies to further improve performance and robustness.

Acknowledgments

I would like to thank the organizers of the AraGenEval shared task and the Arabic NLP community for providing this valuable opportunity and platform for collaboration. Special thanks to everyone who contributed and shared their solutions to

[4] https://www.kaggle.com/code/thameralharbi/arageneval-subtask3-aratect

[5] https://www.codabench.org/competitions/8545/#/results-tab

[6] https://www.codabench.org/competitions/9120/#/results-tab

help others and advance the community. Finally, I am deeply grateful to my parents for their continuous support and encouragement throughout this journey.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Association for Computational Linguistics.

Maged S Al-Shaibani and Moataz Ahmed. 2025. The arabic ai fingerprint: Stylometric analysis and detection of large language models text. *arXiv preprint arXiv:2505.23276*.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham Abdullah Alyahya, Sultan AlRashed, Faisal Abdulrahman Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Saad Amin Hassan, Dr. Majed Alrubaian, Ali Alammari, Zaki Alawami, and 7 others. 2025. ALLam: Large language models for arabic and english. In *The Thirteenth International Conference on Learning Representations*.

NAMAA. 2025. Aramodernbert: Advanced arabic language model through trans-tokenization and modernbert architecture. `https://huggingface.co/NAMAA-Space/AraModernBert-Base-V1.0`. Accessed: 2025-03-02.

Nojoom.AI at AraGenEval Shared Task: Arabic Authorship Style Transfer

Hafsa Kara Achira
Nojoom.AI
ih_karaachira@esi.dz

Mourad Bouache
Nojoom.AI
bouache@nojoom.ai

Mourad Dahmane
Nojoom.AI
mdahmane@gmail.com

Abstract

This paper presents our approach and findings for Subtask 1 (Authorship Style Transfer) of the AraGenEval2025 shared task. We explore methods to transform neutral Arabic text into the distinctive style of a specified author while preserving its original meaning. Our work details a two-phase development: an initial baseline model leveraging few-shot prompting with Gemini and K-means clustering, followed by fine-tuning of pre-trained seq2seq models that support Arabic, including representatives from the mT5 and mBART model families. We evaluated our models using BLEU and chrF metrics, demonstrating significant improvements in fine-tuning, particularly in capturing Arabic-specific stylistic nuances. To complement these surface-level overlap metrics, we incorporate BERTScore to assess semantic preservation across style transfer. Additionally, we introduce a style classifier to quantify author-specific style transfer strength. We discuss the challenges encountered, including Arabic linguistic complexities, handling long Arabic text, and hardware constraints, and outline future directions for enhancing Arabic Authorship Style Transfer.

1 Introduction

The proliferation of digital content requires advanced natural language processing (NLP) techniques for text manipulation. Authorship style transfer (AST) is a challenging yet key task aiming to convert a given text into the writing style of a target author while maintaining its semantic content. This differs from traditional stylistic analysis, focusing solely on identifying and characterizing an author's style. The increasing sophistication of AI-generated content, particularly in Arabic, further highlights the need for robust AST models, as style identification can contribute to detecting synthetic texts.

Despite its importance, Arabic AST remains a relatively underexplored area compared to other languages. The Arabic language presents unique linguistic challenges, including significant morphological variations, rich affixation, diverse dialects, and complex reordering phenomena, all of which impact style transfer. Furthermore, the scarcity of large-scale labeled datasets for Arabic AST poses a significant hurdle. This complexity is further exacerbated by the high inflectional nature of Arabic, which introduces tokenization difficulties, especially when dealing with long texts and paragraph-level inputs.

The AraGenEval2025 shared task, hosted with the Arabic Natural Language Processing (ArabicNLP 2025) Conference (Abudalfa et al., 2025), aims to foster research in this domain. Our participation focuses on Subtask 1: Authorship Style Transfer, where the objective is to transform a formal input text into a specified author's style. This paper details our methodology, experimental setup, evaluation, and the insights gained throughout the project, and concludes with perspectives for future works.

Our system entails a two-stage strategy: an initial baseline using few-shot prompting with *Gemini*, supported by K-means clustering, followed by fine-tuning of Arabic-supporting seq2seq models from the *mT5*, *AraT5*, and *mBART* families. The resulting models achieved **24.46%** and **59.33%** in BLEU and chrF, respectively, reflecting word- and character-level surface overlap with reference texts. Meaning preservation across style transfer was measured at **92.01%** using *BERTScore*. The stylistization precision per author reached **86.12%**, as assessed using the style classifier. Implementation is available at[1].

2 Background

While Arabic AST remains relatively underexplored, two recent approaches (Shao et al., 2024) and (Hu et al., 2022) provide valuable foundations. Both generate pseudo-parallel neutral↔stylized pairs using GPT and fine-tune a seq2seq model on sentence-level data. (Shao et al., 2024) focuses on general purpose style transfer and has been applied to well-defined styles such as Shakespeare, rap lyrics, and Chinese literature. It leverages

[1] https://github.com/nojoom-ai/AraGenEval2025

English- and Chinese-centric tokenizers and pre-trained BART models. Stylized samples are selected using K-means clustering and augmented bidirectionally to train a BART-based model.(Hu et al., 2022), on the contrary, targets a few-shot style transfer with low-resource authors. It applies GPT-based neutralization followed by supervised fine-tuning and introduces a reward model to guide output refinement through preference-based policy optimization.

Despite their strengths, both approaches are limited to short-form inputs, rely heavily on English-centric infrastructure, and employ evaluation setups that do not account for Arabic's morphological complexity or long-form stylistic variation. Our work addresses these limitations by extending AST to paragraph-level Arabic inputs, explicitly managing tokenization challenges caused by high inflection. We fine-tune Arabic-supporting seq2seq models and propose a broader evaluation protocol, inspired by (Shao et al., 2024) and (Hu et al., 2022).

3 System Overview

The system comprises two stages, inspired by the (Shao et al., 2024) and (Hu et al., 2022) approaches. First, we develop a baseline model that serves as a reference for comparison (see Fig. 2). Next, we fine-tune several Arabic-supporting pre-trained models.

3.1 Baseline Model

Our initial approach utilizes few-shot prompting with *Gemini 2.5 Flash*. The process involves:

- **K-means Clustering** (Fig. 2 Step 1.a): We performed exploratory data analysis (EDA) on embedding representations of training samples, using the elbow method and silhouette scores to determine that $k = 2 – 3$ clusters are optimal for most authors. We then applied K-means to select the top $K = 3$ representative examples per author.

- **Prompt Construction**: We construct a prompt by concatenating the selected exemplars with the neutral input text.

- **Styled Output Generation** (Fig. 2 Step 1.b): *Gemini 2.5 Flash* generates the stylized output based on the constructed prompt.

3.2 Pre-trained Models Fine-Tuning

To address the limitations of the few-shot baseline, we implemented a fine-tuning pipeline for pre-trained seq2seq models (phase 2):

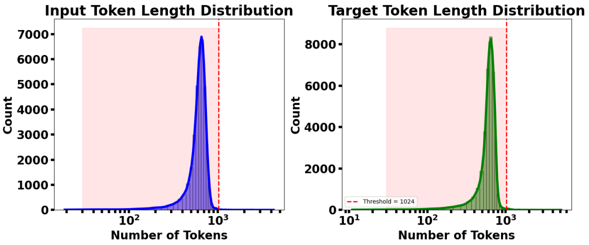


Figure 1: Token-length distributions for training dataset input (blue) and target (green).

%	In	Tgt	%	In	Tgt
0	19	11	95	781	765
5	433	419	98	822	825
11	509	501	99	870	934
50	644	635	Q3+1.5·IQR	877	864
90	748	735	100	4248	5094

Table 1: Training Set Input and target token-length statistics. Q3+1.5·IQR indicates the statistical outlier upper threshold.

- **Input Preparation**: For each training sample, we prepend an author tag to the neutral text. The corresponding stylized text is used as the target sequence.

- **Tokenization** (Fig. 2 Step 2.a): Arabic morphology is highly inflected and rich in prefixes and suffixes, resulting in a higher subword token count per word compared to English. (Rust et al., 2021) shows that Arabic typically yields 1.1–1.8 subword tokens per word, compared to 1.2–1.3 in English. Since VRAM usage scales roughly with the square of sequence length, we selected our token-length caps to balance dataset coverage and hardware constraints.

  We analyze token-length distributions across training and validation sets using the *mBART50* tokenizer (Fig. 1). A maximum length of 750 tokens covers $\approx$ 90% of the samples, while 1024 tokens cover $\approx$ 99.6% (see Table 1). The final tokenization limits were chosen based on the available hardware and pre-trained model sizes.

- **Fine-Tuning** (Fig. 2 Step 2.b): The pre-trained model weights (*mT5*, *AraT5*, *mBART*) were fine-tuned on the prepared dataset, with intermediate checkpoints saved to handle long training sessions.

- **LoRA Injection** (Fig. 2 Step 2.c): To improve

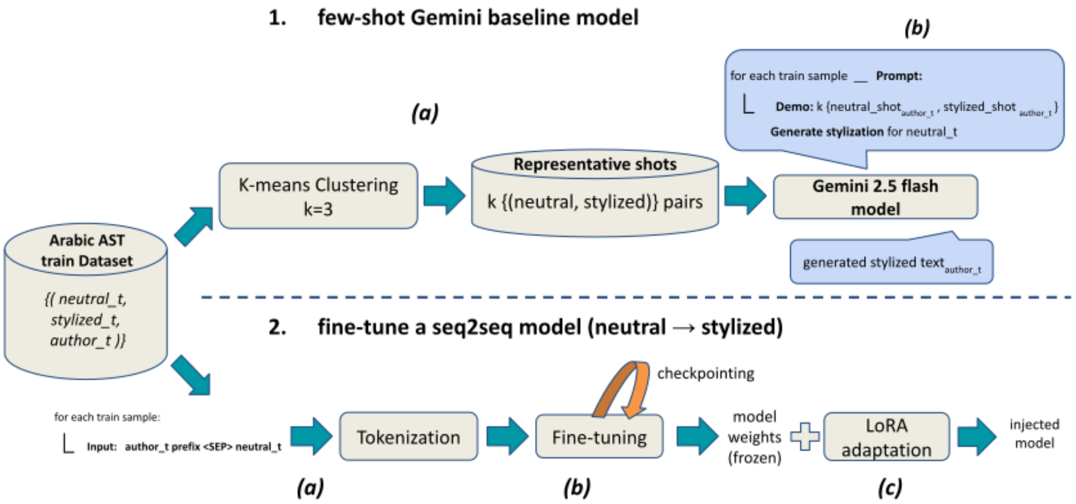

1. few-shot Gemini baseline model
(b)
for each train sample __ Prompt:
Demo: k {neutral_shot author_t , stylized_shot author_t }
Generate stylization for neutral_t
(a)
K-means Clustering k=3
Representative shots
k {(neutral, stylized)} pairs
Gemini 2.5 flash model
generated stylized text author_t
Arabic AST train Dataset
{(neutral_t, stylized_t, author_t)}
2. fine-tune a seq2seq model (neutral → stylized)
checkpointing
for each train sample:
Input: author_t prefix <SEP> neutral_t
Tokenization
Fine-tuning
model weights (frozen)
LoRA adaptation
injected model
(a)
(b)
(c)

Figure 2: Arabic AST Model developement pipeline

performance under hardware constraints, we injected Low-Rank Adaptation (LoRA) modules (Hu et al., 2021) into the fine-tuned models and conducted additional training on the training dataset.This enabled further optimization over more epochs while keeping the base model weights frozen.

4 Experimental Setup

4.1 Data Splits

We use the official *AraGenEval2025* dataset, consisting of 35,122 paragraph-level samples for training (72.1%), 4,157 for validation (8.5%), and 8,143 for testing (19.3%). The test set labels are withheld by the organizers and used only for final evaluation. Tokenized input lengths reach up to 3,361 tokens, with 99.66% of samples under 1,024 tokens (Fig. 4).

4.2 Preprocessing

Each neutral input is prepended with an author tag in the format: `<AUTHOR> | <NEUTRAL_TEXT>`. Tokenization is performed using the corresponding *AutoTokenizer* for each model.

4.3 Hardware and Environments

All experiments were conducted on cloud-based platforms with varying GPU configurations; full details are provided in Appendix B.1.

4.4 Evaluation Metrics

We report the two official competition metrics - **BLEU** and **chrF** to assess word- and character-level surface overlap. In addition, we include two complementary metrics: **BERTScore (BS)**, for measuring semantic preservation, and **Style Classifier Accuracy (SC)**, to assess author-specific style transfer strength. Details of the style classifier development are provided in Appendix B.6.

5 Results

This section presents the empirical evaluation of our AST models, detailing their performance across various metrics, and providing per-author insights. Our models were evaluated on validation dataset. The best performing models were then used on the final test data set evaluation.

5.1 Overall Performance Comparison

Table 5 summarizes the performance of the Few-Shot baseline and various fine-tuned models. Overall, fine-tuning yields substantial gains: BLEU improves from 11.66 to 24.46 ($\Delta = +11.26$) and chrF from 48.12 to 59.33 ($\Delta = +11.21$), confirming improved stylistic alignment. BS remains consistently high ($\sim$ 0.91–0.93), indicating strong meaning preservation across models. SC aligns well with other metrics, supporting its usefulness in quantifying stylistic strength.

Among the models evaluated, *Facebook/mbart-large-mmt-50* attains the highest validation BLEU and chrF, while `UBC-NLP/AraT5-v2-1024` is highly competitive in both validation and test results given its parameter weight. LoRA injection on `UBC-NLP/AraT5-v2-1024` yielded modest gains where applied; overall improvements are primarily attributable to fine-tuning.

Although the gains are clear, chrF scores in the high 50s suggest remaining challenges in capturing

Validation Set Results				
Model	**BLEU**	**chrF**	**BS**	**SC**
Few-Shot Baseline	11.66	48.12	91.25	58.43
google/mt5-small	18.51	52.92	91.88	59.78
UBC-NLP/AraT5-base	21.24	57.13	92.02	62.20
agemagician/mlong-t5-tglobal-large	23.58	58.88	93.01	73.58
facebook/mbart-large-mmt-50	**24.56**	**59.92**	92.01	**85.86**
moussakam/AraBART	21.76	58.21	92.52	58.67
UBC-NLP/AraT5-v2-1024	23.80	59.27	91.63	73.90

Table 2: Validation set results for evaluated models.

Test Set Results			
Model	**BLEU**	**chrF**	**SC**
facebook/mbart-large-50	24.46	59.33	86.18
moussakam/AraBART	21.07	57.21	59.12
UBC-NLP/AraT5-v2-1024	24.07	59.48	74.31
UBC-NLP/AraT5-v2-1024 **+ LoRA**	24.22	59.53	75.42

Table 3: Test set results for selected models. LoRA was injected only where indicated.

Author	**Cnt**	**BLEU**			**chrF**			**Author**	**Cnt**	**BLEU**			**chrF**		
		B	**FT**	**Δ**	**B**	**FT**	**Δ**			**B**	**FT**	**Δ**	**B**	**FT**	**Δ**
A. G. Makawi	396	17.16	31.48	+14.32	55.07	66.64	+11.57	Ahmed Amin	246	9.47	18.77	+9.30	47.09	57.09	+10.00
Fouad Zakaria	125	17.10	27.02	+9.92	54.27	62.62	+8.35	A. M. Al-Aqqad	267	8.67	17.89	+9.22	44.76	54.15	+9.39
Naguib Mahfouz	327	15.21	25.49	+10.28	50.66	59.60	+8.94	Salama Moussa	119	8.05	14.53	+6.48	44.51	53.95	+9.44
Jurji Zaydan	327	14.39	21.48	+7.09	52.24	59.15	+6.91	Yusuf Idris	120	7.48	17.71	+10.23	42.79	55.08	+12.29
Robert Bar	82	13.75	19.16	+5.41	49.90	54.02	+4.12	**G. K. Gibran**	30	7.18	27.87	+20.69	45.35	61.44	+16.09
Tharwat Abaza	90	12.96	27.71	+14.75	50.15	59.93	+9.78	M. H. Heikal	260	6.07	14.31	+8.24	42.84	52.21	+9.37
Hassan Hanafi	548	12.93	25.04	+12.11	48.59	61.20	+12.61	Taha Hussein	255	5.68	14.54	+8.86	42.12	51.59	+9.47
Amin Al-Rihani	142	12.65	21.62	+8.97	51.12	59.93	+8.81	A. Teimur Pasha	57	3.76	17.74	+13.98	30.53	46.39	+15.86
W. Shakespeare	238	11.35	26.21	+14.86	48.08	61.02	+12.94	Kamel Kilani	25	2.43	13.38	+10.95	34.03	50.64	+16.61
N. El Saadawi	295	10.83	29.77	+18.94	48.28	65.90	+17.62	**Ahmed Shawqi**	58	1.91	19.34	+17.43	37.72	55.49	+17.77
Gustave Le Bon	150	9.60	18.60	+9.00	48.96	59.05	+10.09	**Overall**	4157	11.66	22.92	+11.26	48.12	59.13	+11.01

Table 4: Per-author performance comparison of the fine-tuned `UBC-NLP/AraT5-v2-1024` vs. the baseline models.

Arabic's morphological richness. These results emphasize the importance of both model architecture and input processing for effective style transfer.

5.2 Per-Author Insights

To gain deeper insights, we analyze per-author performance of the fine-tuned *UBC-NLP/AraT5-v2-base-1024* model (367M parameters) against the baseline. We chose it because of its strong performance compared to *mBART-large-50-mmt* at lower parameter cost, and because it better handles long inputs (full-sample tokenization); see Appendix A and §B.1. Table 4 reports BLEU and chrF per author with absolute changes (Δ).

The analysis shows consistent gains across most authors. Notable examples include *Gibran Khalil Gibran* (30 samples), which exhibits the largest increase ($\Delta_{\text{BLEU}} = +20.69$, $\Delta_{\text{chrF}} = +16.09$); *Ahmed Shawqi* (58 samples) also shows strong improvements (+17.43, +17.77); and *Nawal El Saadawi* (295 samples) with substantial gains (+18.94, +17.62). Overall, the model achieves a sizable overall uplift (BLEU ↑ 11.26, chrF ↑ 11.01), demonstrating that AraT5-v2-1024 effectively captures author-specific stylistic signals while handling longer inputs, and may surpass the model *mBART-large-50-mmt*, if a considerable share of long inputs ($>$ 1024 tokens) were present in the evaluation sets.

Conclusion

Our participation in Subtask 1 of AraGenEval2025 demonstrates effective Authorship Style Transfer for Arabic. Building on a few-shot Gemini 2.5 Flash with shots selection through K-means clustering baseline, we fine-tuned arabic-supporting seq2seq models, achieving 24.46% BLEU, 59.33% chrF, 92.3% BS and 86% SC. Per-author results were consistently strong, with the lightweight *UBC-NLP/AraT5-v2-1024* (367 M parameters) matching or exceeding larger multilingual models, underscoring the value of Arabic-specific pre-training.

We identified several Arabic AST challenges , including rich morphology and affixation, dialectal variation, reordering, and long paragraph inputs. We tackled long training on limited hardware by injecting LoRA modules and using token-budgeted batching with CPU/GPU overlap to respect hardware limits while processing extended contexts.

Although chrF improvements indicate further room for capturing fine-grained character-level nuances, our approach lays a solid foundation. Future work will explore longer inputs handling, and integrate human-in-the-loop evaluation (e.g., Gemini judgment) to further enhance stylistic fidelity.

6 Acknowledgments

The authors gratefully acknowledge the invaluable support and resources provided by the ***Nojoom.AI*** team. Their dedication, technical guidance, and infrastructure were instrumental in the successful execution of this research and our participation in the *AraGenEval2025* shared task. We extend our sincere gratitude for their continuous encouragement and collaborative spirit.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Association for Computational Linguistics.

Moussa Kamal Eddine, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. 2022. Arabart: a pretrained arabic sequence-to-sequence model for abstractive summarization. arXiv preprint arXiv:2203.10945.

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 ArabicNLP 2023*, pages 232–243, Singapore (Hybrid). Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint*, arXiv:2106.09685.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal, and Aston Zhang. 2022. Text style transfer: A review and experimental evaluation. *ACM SIGKDD Explorations Newsletter*, 24(1):14–45.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. Arat5: Text-to-text transformers for arabic language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 628–647.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Zhonghui Shao, Jing Zhang, Haoyang Li, Xinmei Huang, Chao Zhou, Yuanchun Wang, Jibing Gong, Cuiping Li, and Hong Chen. 2024. Authorship style transfer with inverse transfer data augmentation. *AI Open*, 5:94–103.

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 finetuning.

David Uthus, Santiago Ontañón, Joshua Ainslie, and Mandy Guo. 2023. mlongt5: A multilingual and efficient text-to-text transformer for longer sequences. arXiv preprint arXiv:2305.11129.

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)*, pages 483–498.

A Appendix: Dataset Distribution Details

In seq2seq tasks, setting an appropriate maximum input length during tokenization is critical for reliable evaluation. Truncating long inputs can degrade performance by removing key information, especially for stylistic tasks that rely on paragraph-level context.

The tables and plots in this appendix provide a detailed overview of the input and target token length distributions for the validation and test sets. These statistics were used to determine safe maximum input lengths that cover at least 99% of the samples, ensuring high coverage without excessive memory consumption. Outlier thresholds based on the $Q3+1.5\cdot IQR$ rule are also reported to highlight extreme cases.

A.1 Validation Set Token-Length Distribution

%	In	%	In
0	21	75	693
5	432	90	736
10	500	95	768
25	574	Q3+1.5·IQR	872
50	639	100	1216

Table 5: Validation set input token-length statistics. Q3+1.5·IQR indicates the statistical outlier upper threshold.

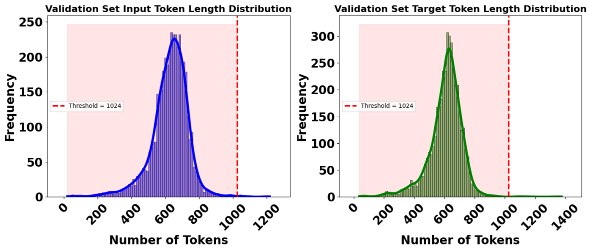


Figure 3: Token-length distributions for validation dataset input (blue) and target (green).

A.2 Test Set Input Token-Length Distribution

It is important to note that different model architectures impose different maximum input length constraints. **mBART-based models** such as `facebook/mbart-large-50-mmt` and `moussakam/AraBART` enforce a *hard limit* of 1,024 tokens due to their absolute positional embeddings. In contrast, **T5-based models** such as `google/mt5-small`, `UBC-NLP/AraT5-base`, and `UBC-NLP/AraT5-v2-1024` utilizes relative positional embeddings, which allow a *soft limit*—they can accept longer sequences as long as the available hardware permits.

%	In	%	In
0	30	75	702
5	433	95	747
10	514	99	855
25	587	Q3+1.5·IQR	877
50	650	100	3361

Table 6: Test set input token-length statistics. Q3+1.5·IQR indicates the statistical outlier upper threshold.

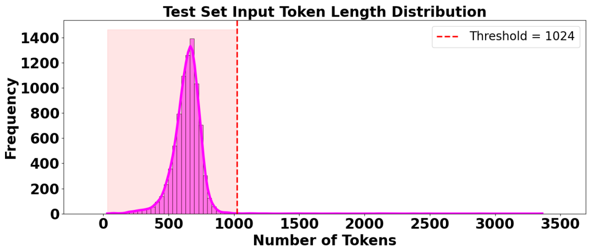


Figure 4: Token length distribution for test set inputs. Over 99.6% of samples fall under 1 024 tokens.

As shown in Table 7, the maximum input lengths used during training and evaluation were configured based on these architectural constraints and the available computing resources. For T5-based models, we set input length limits to 750 or 1,024 tokens to safely cover most validation and test samples without truncation.

B Appendix: Experimental Details

B.1 Model and Environment Details

Table 7 summarizes the models used, their parameter sizes, token length limits, and compute environments. T5-based models tolerate flexible input lengths (hardware permitting), while mBART-based models impose a strict 1024-token cap. Training was conducted on either Colab Pro+ (A100) or Kaggle (P100). CPU-only runs were reserved for small-scale evaluation like ChrF , BLEU and BERTScore calculations due to memory limitations.

B.2 K-means Clustering for Few-Shot Samples Selection

To avoid suboptimal or noisy few-shot examples resulting from random selection, we apply clustering

Model	Params	Platform	Accel.	Training		Evaluation Max tok.	
				unit BS	Max tok.	Validation	Test
`google/mt5-small` (Xue et al., 2021)	310M	Kaggle	P100	1	750	750	/
`UBC-NLP/AraT5-base`(Nagoudi et al., 2022)	280M	Kaggle	P100	1	750	1500	/
`agemagician/mlong-t5-tglobal-large` (Uthus et al., 2023)	1768M	Colab Pro+	A100	4	1024	1500	/
`facebook/mbart-large-50-mmt` (Tang et al., 2020)	610M	Colab Pro+	A100	8	1024	1024	1024
`moussakam/AraBART` (Eddine et al., 2022)	139M	Kaggle	P100	16	1024	1024	1024
`UBC-NLP/AraT5-v2-1024` (Elmadany et al., 2023)	367M	Colab Pro+	A100	12	1024	1500	3500

Table 7: Compute platforms and sequence-length configurations across dataset splits.

of K-means on sentence embeddings to deterministically select representative neutral samples per author. The goal is to ensure that stylistically central examples are used in prompt-based evaluation, without model fine-tuning.

We encode each author's neutral training texts using the `all-MiniLM-L6-v2` model, then cluster the resulting embeddings and extract the closest samples to each cluster centroid as the selected few-shot examples.

Parameter	Value / Setting
Embedding model	`all-MiniLM-L6-v2`
Embedding dimension	384
Clustering method	K-means (per author)
Number of clusters (k)	3
Distance metric	Euclidean
Selection criterion	Centroid-nearest samples
Random seed	42

Table 8: K-means clustering setup for representative few-shot selection.

B.3 Training Configuration

Key hyperparameters (defaults unless otherwise noted):

Parameter	Value
Effective batch size	32
Gradient accumulation steps	8
Max sequence length	750 / 1024
Checkpoint interval	500 steps
Epochs	3
Optimizer	AdamW
Learning rate	5×10^{-5}

Table 9: Summary of training hyperparameters.

B.4 Evaluation Configuration

Inference is performed via a single-GPU, token-budgeted batching pipeline that overlaps CPU tokenization with GPU generation to maximize throughput and avoid OOMs. Inputs are first sorted by length on the CPU, grouped into batches whose total token count does not exceed a configurable budget, then transferred to the GPU for generation. If an OOM occurs, the budget is halved and the batch is retried in smaller splits.

Key parameters are summarized in Table 10.

Parameter	Value / Description
Token budget	10 000 total input tokens
VRAM Memory threshold	80 % of GPU VRAM
Budget increment	+1 000 tokens when VRAM<VRAM_THRESH
Budget update frequency	every 5 successful batches
Max input length	3 400 tokens (capped by model input handling)
Max generation length	4 000 tokens (capped by model input handling)

Table 10: Key settings for token-budgeted inference

This setup ensures that: (1) very long inputs are safely handled without silent truncation, and (2) GPU utilization remains high by feeding pre-tokenized batches as soon as memory permits.

B.5 LoRA Configuration

To enable lightweight and fast adaptation over limited resources, LoRA was injected into attention layers of a frozen *UBC-NLP/AraT5-v2-1024* base. This setup drastically reduces trainable parameters, making hyperparameter sweeps and multi-run experimentation feasible within constrained GPU environments. We used an aggressive injection configuration with moderately high rank and scaling

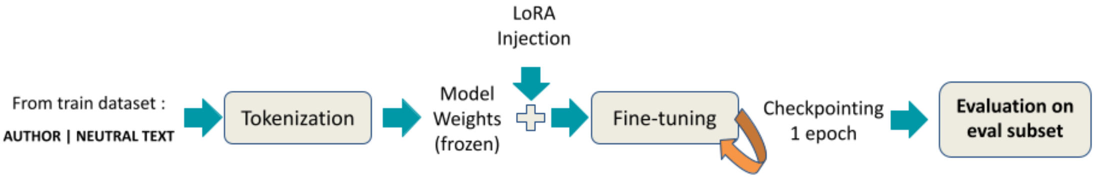


Figure 5: LoRA injection Development & Evaluation pipeline

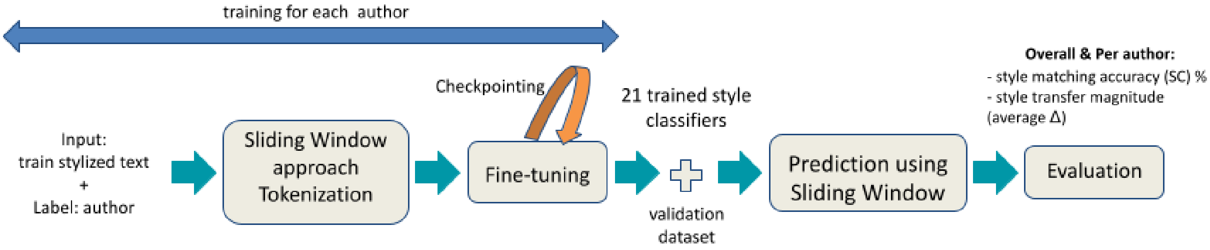

training for each author
Input:
train stylized text
+
Label: author
Sliding Window approach Tokenization
Checkpointing
Fine-tuning
21 trained style classifiers
validation dataset
Prediction using Sliding Window
Evaluation
Overall & Per author:
- style matching accuracy (SC) %
- style transfer magnitude (average Δ)

Figure 6: Arabic Style classifier Development & Evaluation pipeline

values. Checkpoints were saved each epoch, and the model with the best chrF score on a held-out validation subset was selected.

Component	Configuration
Base model	`UBC-NLP/AraT5-v2-1024`
Target modules	`q, v, k, fc1, fc2`
Injection layers	Encoder and decoder attention
Rank (r)	32
Scaling factor (α)	64
Dropout	0.1
Bias	None
Epochs	5
Eval subset	25% subset stratified from validation set
Checkpointing	Every epoch
Final model selection	Best checkpoint by chrF

Table 11: Summary of LoRA fine-tuning configuration.

B.6 Style Classifier

While BLEU and chrF quantify surface overlap, they do not directly measure whether the generated text truly mirrors an author's stylistic fingerprint. To address this, we train an author-specific binary classifier, based on `bert-base-arabic-camelbert-ca`, that learns the distinctive phrasing, vocabulary, and structural patterns of each author.

Unfortunately, no off-the-shelf Arabic style classifier supports long inputs beyond 512 tokens. Our options were to pre-train an English long-document model (e.g. Longformer) on Arabic data or to adopt a sliding-window approach. As shown in Fig. 6, we chose the latter: inputs are split into overlapping 512-token chunks (256-token stride), each classified separately, and results are aggregated. This ensures we capture stylistic cues from long paragraphs without truncation.

For each sample evaluated (from validation or test datasets), we compare the confidence of the classifier in the 'Author X' class on the neutral input versus the stylized output to calculate

$$\Delta = p_{\text{out}}(1) \;-\; p_{\text{in}}(1).$$

An instance is a *hit* if $\Delta > 0$, i.e. the generated output aligns more with the 'Author X' style than with the neutral text (that is,a successful style transfer). We report the hit rate as the SC metric.

Parameter	Setting
Base model	`bert-base-arabic-camelbert-ca`
Input length limit	512 tokens (sliding window)
overlap	256 tokens between chunks
Training epochs	5
Batch size	16
Learning rate	2×10^{-5}
Optimizer	AdamW
Scheduler	Linear warmup
Loss	Binary cross-entropy
Output metrics	Hit rate ($\Delta > 0$), mean Δ
Classifiers	One per author (21 total)

Table 12: Training Setup for each author style classifiers.

Future work should explore pre-training or adapting a native Arabic long-input classifier, rather than relying on sliding windows, to more seamlessly handle long input LLM generations evaluation.

LMSA at AraGenEval Shared Task: Ensemble-Based Detection of AI-Generated Arabic Text Using Multilingual and Arabic-Specific Models

Kaoutar Zita[1]*, Attia Nehar[2], Abdelkader Khelil[2], Slimane Bellaouar[1], Hadda Cherroun[3]

[1]Laboratoire des Mathématiques et Sciences Appliquées (LMSA), Université de Ghardaia, Algeria
[2]Faculty of Exact Sciences and Computer Science, University of Djelfa, Algeria
[3]Laboratoire d'informatique et des Mathématiques, Université Amar Telidji, Laghouat, Algeria
{zita.kaoutar, bellaouar.slimaneg}@univ-ghardaia.edu.dz,
{neharattia, a.khelil}@univ-djelfa.dz,
hadda.cherroun@lagh-univ.dz

Abstract

We address the problem of distinguishing between human-authored and AI-generated text in low-resource languages, particularly Arabic. We present the LMSA[1] team's participation in the ARATECT (Arabic AI-Generated Text Detection) subtask of the AraGenEval[2] shared task, which targets the detection of AI-generated Arabic texts. We propose an ensemble-based classification framework that integrates multilingual and Arabic-specific pre-trained language models, namely Fanar, AraBERT, and XLM-R, optimized through a dedicated fine-tuning pipeline. The approach is evaluated on the balanced Arabic text dataset provided by the shared task organizers. Our system achieved an F1-score of 0.864 and ranked first among all participating teams.

1 Introduction

The rapid advancement of generative artificial intelligence has significantly transformed the landscape of content creation, education, and communication. State-of-the-art large language models (LLMs) such as GPT-3 (Brown et al., 2020), GPT-4 (OpenAI et al., 2023), and LLaMA (Touvron et al., 2023) are now capable of producing text that exhibits a high degree of fluency, coherence, and stylistic refinement, often closely resembling human writing. These technologies offer substantial benefits, including personalized learning, writing support, and scalable content generation. However, they also raise serious ethical concerns regarding authorship, originality, and academic integrity. Moreover, generative AI can be misused to produce misleading or deceptive content, including fabricated news articles (Ishraquzzaman et al., 2025), deepfake tweets (Fagni et al., 2021), and AI-generated documents such as academic papers and study reports (Chowdhury et al., 2025). Such misuse poses significant ethical risks across domains, including journalism, education, and social media. In light of these developments, there is a growing need and a corresponding challenge to reliably distinguish between human-written and machine-generated text.

Arabic, one of the six official languages recognized worldwide (Wahdan et al., 2020) and the fourth most used language on the Internet with over 400 million speakers (Guellil et al., 2021), has received comparatively less attention in the area of AI-generated text detection. In this context, the AraGenEval shared task (Arabic Authorship Style Transfer and AI-Generated Text Detection) (Abudalfa et al., 2025) is introduced to foster research on Arabic text generation and detection. One of its subtasks, ARATECT, focuses on the binary classification of Arabic texts as either human-written or AI-generated.

To address this challenge, we propose an ensemble-based classification framework that combines the strengths of both multilingual and Arabic-specific pre-trained language models. By integrating Fanar, AraBERT, and XLM-R within a fine-tuning pipeline and applying a majority voting strategy, this approach enhanced the robustness and accuracy of our system, enabling it to rank first among the 16 submitted systems in the ARATECT subtask.

The implementation is publicly available[3] to support transparency and reproducibility.

*Corresponding author: zita.kaoutar@univ-ghardaia.edu.dz

[1] Laboratoire des Mathématiques et Sciences Appliquées, University of Ghardaia, Algeria

[2] https://ezzini.github.io/AraGenEval/

[3] https://github.com/kaoutarzi/AraGenEval-2025-Aratect

Proceedings of The Third Arabic Natural Language Processing Conference, pages 26–31
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

2.1 Task Setup

In this study, we address the detection of AI-generated Arabic text as part of the ARATECT subtask in the AraGenEval Shared Task. This subtask is formulated as a binary classification problem in which the system is given an Arabic text and must determine whether it was written by a human or generated by an AI model. The dataset used consists of Arabic texts spanning various genres, including news articles and literary content. It is balanced in terms of class distribution, featuring an equal number of human- and machine-generated samples. The full dataset comprises 5,798 texts, split into training, development, and test sets, as detailed in Table 1.

For instance, a system might encounter a news excerpt such as:

" قالت وكالة الأنباء السورية سانا إن الدفاعات الجوية السورية تصدت لعدوان إسرائيلي بعدد من الصواريخ استهدفت مناطق في محيط العاصمة دمشق في الساعات الأولى من اليوم الخميس ."

Which means "The Syrian Arab News Agency (SANA) reported that Syrian air defenses responded to an Israeli attack involving several missiles that targeted areas around the capital, Damascus, in the early hours of Thursday." The system is then expected to classify the text accordingly.

2.2 Related Work

Numerous studies (Liu et al., 2025; Wu et al., 2025; Fraser et al., 2025) have addressed the challenge of detecting AI-generated text, driven by the growing capabilities of large language models. However, most existing research has focused predominantly on English or other high-resource languages.

For instance, Katib et al. (2023) introduced a hybrid model called TSA-LSTMRNN, which integrates LSTM with an attention mechanism and the Tunicate Swarm Algorithm (Kaur et al., 2020). They utilize TF-IDF, count vectorizer, and word embeddings for feature extraction, achieving up to 93.83% accuracy in distinguishing between human- and ChatGPT-generated text.

Antoun et al. (2023) proposed a methodology for detecting ChatGPT-generated French text by translating the HC3 English dataset (Guo et al., 2023) and training classifiers (e.g., CamemBERTa, XLM-R). The detectors performed well in-domain (F1 $\approx$ 0.97), but showed reduced effectiveness on out-of-domain and adversarial samples, highlighting limitations in generalization.

Focusing specifically on Arabic, Alshammari et al. (2024) propose two fine-tuned Transformer-based models, AraELECTRA and XLM-R, for detecting AI-generated versus human-written texts. Their approach incorporates a novel Dediacritization Layer. Trained on the AIRABIC dataset (Alshammari and EI-Sayed, 2023), the models achieve up to 83% accuracy, outperforming GPTZero (63%) and OpenAI Text Classifier (50%).

Similarly, Alghamdi and Alowibdi (2024) compiled a dataset of Arabic tweets authored by both humans and ChatGPT. They trained and evaluated three machine learning models (SVM, Naive Bayes, and Decision Tree), with Naive Bayes achieving the highest accuracy of 93% in distinguishing between the two sources.

3 System Overview

In this study, we progressively explored a wide range of models for Arabic text classification to address the task of detecting AI-generated content. We began with traditional machine learning methods, advanced through deep learning architectures, and further extended our investigation by fine-tuning various pre-trained language models. To enhance overall performance and robustness, we adopt an ensemble strategy based on majority voting (Dong et al., 2020). The following sections provide a detailed exploration of each category of models employed in our study.

3.1 Machine Learning-based Classification

To classify Arabic AI-generated text using traditional machine learning, we extracted three types of features: (1) statistical and stylistic features, such as word counts, lexical diversity, and punctuation usage; (2) TF-IDF features, which captured sparse lexical patterns; and (3) contextual representations derived from AraBERT embeddings. These features were then used as input to machine learning models, specifically Logistic Regression and a Multi-Layer Perceptron (MLP), which were selected based on their performance on the development set.

3.2 Deep Learning-based Classification

To explore deep learning-based detection, we designed a fusion architecture that integrates both handcrafted and contextual features. As shown in Figure 1, the input text is processed twice to get a rich encoding. The first branch encodes handcrafted stylometric and sparse lexical patterns (stylistic features and TF-IDF), while the second processes semantic features obtained via AraBERT embeddings. This separation aims to preserve the distinct contribution of each feature type and prevent potential dominance of contextual embeddings. The outputs from both branches are then concatenated and passed through a multi-head attention layer to model cross-feature interactions, enabling the integration of both surface-level and deep contextual cues for the final classification.

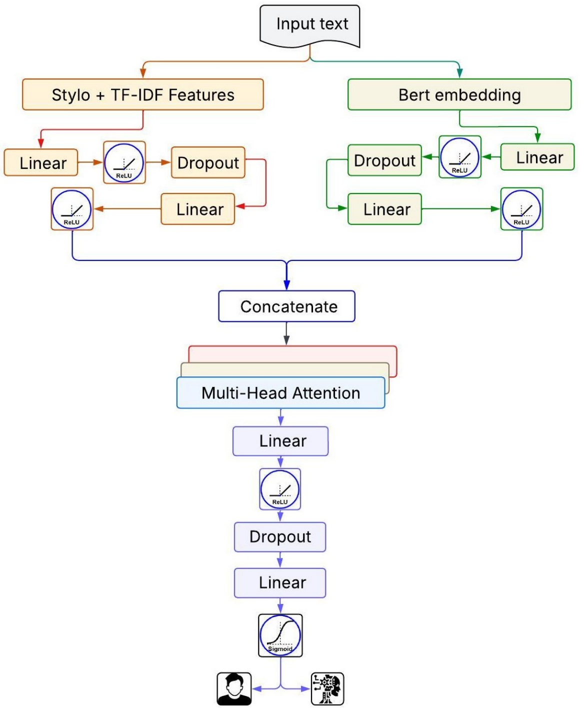


Figure 1: FusionNet Architecture for Arabic AI Generated Text Detection.

3.3 LLM-based Classification

A core focus of our work lies in exploring the potential of large pre-trained language models (LLMs) for detecting AI-generated text. To this end, we experimented with several models and identified three that contributed the most significantly to our final submission results: Fanar, AraBERT, and XLM-R.

XLM-RoBERTa[4] is a multilingual transformer-based language model developed to handle over 100 languages, including Arabic. It builds upon the RoBERTa architecture and is trained using the Masked Language Modeling (MLM) objective on a massive dataset of 2.5TB of filtered CommonCrawl data. Its architecture supports fine-tuning for tasks such as text classification, sentiment analysis, and question answering, leveraging rich contextual representations learned from diverse multilingual corpora (Conneau et al., 2020).

AraBERT[5] is a transformer-based language model specifically pre-trained for Arabic, adapting the original BERT (Devlin et al., 2019) architecture to better address the linguistic richness and morphological complexity of Arabic. Trained on approximately 1.5 billion words from diverse Arabic corpora, AraBERT demonstrates strong performance across various NLP tasks such as sentiment analysis, question answering, and named entity recognition. Its design, which includes 12 encoder layers and 136M parameters, allows it to capture deep contextual representations tailored to the Arabic language (Antoun et al., 2020).

Fanar[6] is an Arabic-centric multimodal Large Language Model developed by the Qatar Computing Research Institute at Hamad Bin Khalifa University. It is available in two versions: Fanar Star (7B) and Fanar Prime (9B), trained on a corpus of one trillion tokens in Arabic and English. Fanar is designed to support Modern Standard Arabic as well as major regional dialects. Aligned with Islamic values and Arab cultural contexts, it offers a range of capabilities such as text generation, speech and image processing, and retrieval-augmented generation (RAG) (Team et al., 2025).

Finally, as shown in Figure 2, the predictions from the fine-tuned XLM-RoBERTa, AraBERT, and Fanar models were combined using a majority voting scheme. This ensemble method leveraged the complementary strengths of the individual models to achieve balanced performance across all evaluation metrics and improve the overall accuracy and robustness of the text classification system.

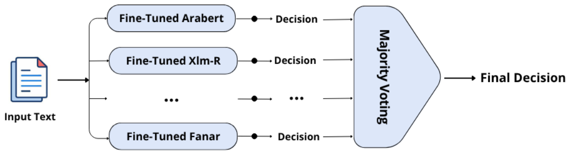


Figure 2: Ensemble-Based Approach for Arabic AI-Generated Text Detection.

4 Experimental Setup

We deploy the dataset provided in the ARATECT subtask of the AraGenEval shared task (Abudalfa

[4] FacebookAI/xlm-roberta-base

[5] aubmindlab/bert-base-arabertv2

[6] https://huggingface.co/QCRI/Fanar-1-9B

et al., 2025), which aims to detect AI-generated Arabic texts. The dataset comprises a balanced set of human- and machine-generated texts across the training, development, and test splits. Human-written texts were sourced from credible Arabic news platforms and literary works, ensuring diversity in style and topic. In contrast, machine-generated texts were produced using multiple large language models, including Mistral, GPT-4, and LLaMA.

Table 1 provides a detailed overview of the dataset's composition.

Data	**Training**	**Dev**	**Test**
# of Samples	4,798	500	500
# of Words	2,330,765	139,745	115,057
Machine (%)	50%	50%	50%
Human (%)	50%	50%	50%

Table 1: ARATECT Dataset Overview.

All experiments were conducted using Python within a Kaggle GPU environment, leveraging the Hugging Face Transformers, Datasets, and Evaluate libraries to fine-tune three pre-trained language models: XLM-RoBERTa, AraBERT, and Fanar. For XLM-RoBERTa and AraBERT, texts were tokenized and classified using cross-entropy loss, with a batch size of 4 over 3 epochs and 1 epoch, respectively. Fanar was fine-tuned using instruction-formatted prompts through LoRA-based parameter-efficient tuning in 4-bit precision, with a batch size of 2 and one epoch. Model performance was evaluated using accuracy, precision, recall, and F1-score. All implementation details, including code and configurations, are publicly available on GitHub[7].

5 Results

Table 2 presents the evaluation results across all experimented models. Traditional machine learning approaches and FusionNet obtained relatively modest performance, reflecting their limited ability to capture the complex linguistic patterns in the dataset. Among the Transformer-based models, the three fine-tuned large language models XLM-R, AraBERT, and Fanar stood out with superior and complementary strengths. AraBERT achieved the highest accuracy (0.864) and F1-score (0.861), XLM-R attained the highest precision (0.911), and Fanar recorded the highest recall (0.920). Although the performance of the Majority Voting ensemble is numerically close to that of AraBERT, the ensemble remains valuable because it balances these strengths, producing a more stable and robust system that is less dependent on the behavior of a single model and better suited to varying data distributions.

Model	**Acc.**	**Prec.**	**Rec.**	**F1**
LR	0.438	0.464	0.804	0.589
MLP	0.506	0.503	0.988	0.667
FusionNet	0.578	0.552	0.824	0.661
AraElectra	0.688	0.737	0.584	0.652
MARBERT	0.586	0.563	0.764	0.649
DeBERTa	0.768	0.791	0.728	0.758
Qwen2.5	0.480	0.490	0.940	0.644
CAMeL	0.642	0.612	0.776	0.684
XLM-R	0.832	0.911	0.736	0.814
AraBERT	0.864	0.882	0.840	0.861
Fanar	0.776	0.714	0.920	0.804
Majority Voting	**0.866**	**0.877**	**0.852**	**0.864**

Table 2: Performance of our models.

6 Conclusion

In this study, we developed a system for AI-generated Arabic text detection within the ARATECT subtask of the AraGenEval Shared Task. We proposed an ensemble-based classification framework that combines the strengths of both multilingual and Arabic-specific pre-trained language models. By integrating Fanar, AraBERT, and XLM-R within a fine-tuning pipeline and applying a majority voting strategy, the system achieved strong and balanced performance across all evaluation metrics. However, there is room for improvement, particularly in enhancing generalization capabilities to unseen domains and handling more diverse writing styles. Future work will address these limitations by exploring more advanced ensemble learning techniques, such as stacking, incorporating larger and more recent language models like GPT-4 or LLaMA 3, and evaluating the system on broader datasets to further improve robustness and adaptability. Furthermore, we plan to extend the classification task beyond binary detection to detect specific AI-generated segments within texts.

[7] https://github.com/kaoutarzi/AraGenEval-2025-Aratect

Acknowledgments

We gratefully acknowledge the organizers of the AraGenEval Shared Task for providing the dataset and evaluation framework used in this study. This work is supported by the Algerian Directorate-General for Scientific Research and Technological Development (DGRSDT), and performed under the PRFU Projects: C00L07N030120220002 and C00L07UN470120230001.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Association for Computational Linguistics.

Noura Saad Alghamdi and Jalal Suliman Alowibdi. 2024. Distinguishing Arabic GenAI-generated Tweets and Human Tweets utilizing Machine Learning. *Engineering, Technology & Applied Science Research*, 14(5):16720–16726.

Hamed Alshammari and Ahmed El-Sayed. 2023. AIRABIC: Arabic Dataset for Performance Evaluation of AI Detectors. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 864–870.

Hamed Alshammari, Ahmed El-Sayed, and Khaled Elleithy. 2024. AI-Generated Text Detector for Arabic Language Using Encoder-Based Transformer Architecture. *Big Data and Cognitive Computing*, 8(3).

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.

Wissam Antoun, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah. 2023. Towards a robust detection of language model-generated text: Is ChatGPT that easy to detect? In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs*, pages 14–27, Paris, France. ATALA.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Shammur Absar Chowdhury, Hind Almerekhi, Mucahid Kutlu, Kaan Efe Keleş, Fatema Ahmad, Tasnim Mohiuddin, George Mikros, and Firoj Alam. 2025. GenAI content detection task 2: AI vs. human – academic essay authenticity challenge. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 323–333, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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.

Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Front. Comput. Sci.*, 14(2):241–258.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):1–16.

Kathleen Fraser, Hillary Dawkins, and Svetlana Kiritchenko. 2025. Detecting AI-Generated Text: Factors Influencing Detectability with Current Methods. *Journal of Artificial Intelligence Research*, 82:2233–2278.

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2021. Arabic natural language processing: An overview. *Journal of King Saud University - Computer and Information Sciences*, 33(5):497–507.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *ArXiv*, abs/2301.07597.

Md Ishraquzzaman, Ashraful Islam, Shahreen Rahman, and Riasat Khan. 2025. Ensemble Transformer-Based Detection of Fake and AI-Generated News.

Applied Computational Intelligence and Soft Computing, 2025.

Iyad Katib, Fatmah Y. Assiri, Hesham A. Abdushkour, Diaa Hamed, and Mahmoud Ragab. 2023. Differentiating Chat Generative Pretrained Transformer from Humans: Detecting ChatGPT-Generated Text and Human Text Using Machine Learning. *Mathematics*, 11(15).

Satnam Kaur, Lalit K. Awasthi, A.L. Sangal, and Gaurav Dhiman. 2020. Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, 90:103541.

Xin Liu, Yang Li, and Kan Li. 2025. Enhancing the Robustness of AI-Generated Text Detectors: A Survey. *Mathematics*, 13(13).

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and Barret Zoph. 2023. GPT-4 Technical Report.

Fanar Team, 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, and 23 others. 2025. Fanar: An Arabic-Centric Multimodal Generative AI Platform. *Preprint*, arXiv:2501.13944.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models.

Ahlam Wahdan, Sendeyah Hantoobi, Said Salloum, and Khaled Shaalan. 2020. A systematic review of text classification research based on deep learning models in Arabic language. *International Journal of Electrical and Computer Engineering (IJECE)*, pages 6629–6643.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Amr&MohamedSabaa at AraGenEval shared task: Arabic Authorship Identification using Term Frequency – Inverse Document Frequency Features with Supervised Machine Learning

Amr Sabaa[1] **and Mohamed Sabaa**[2]
[1]Department of Biomedical Engineering, Cairo University, Giza, Egypt
[2]Department of Computer Science, Najran University, Najran, Saudi Arabia
amr.said01@eng-st.cu.edu.eg, 444307237@nu.edu.sa

Abstract

This paper presents our approach to the AraGenEval 2025 shared task on Arabic authorship attribution (Task 2). We developed an enhanced traditional machine learning system that combines word-level and character-level TF-IDF features with multiple classification algorithms. Our system achieved 88.90% accuracy and 82.74% macro F1-score on the official test set using Logistic Regression. During development, we evaluated multiple models on the validation set, where Linear SVM achieved the highest performance with 93.22% accuracy and 87.52% macro F1-score. The approach demonstrates the effectiveness of feature engineering and proper text preprocessing for Arabic authorship attribution tasks without relying on deep learning architectures.

1 Introduction

Authorship attribution is a fundamental task in computational linguistics that aims to identify the author of a given text based on stylistic patterns and linguistic features (Stamatatos, 2009). For Arabic texts, this task presents unique challenges due to the language's morphological complexity, rich orthographic variations, and diverse dialectal forms.

The AraGenEval 2025 shared task on Arabic authorship attribution (Abudalfa et al., 2025) provides a benchmark for evaluating computational approaches to identifying authors from a collection of Arabic literary texts. This task is particularly relevant in digital humanities, forensic linguistics, and plagiarism detection for Arabic content.

Our contribution focuses on developing a robust traditional machine learning approach that leverages carefully engineered features and proven classification algorithms. We present a comprehensive preprocessing pipeline specifically designed for Arabic literary texts, an effective combination of word-level and character-level Term Frequency - Inverse Document Frequency (TF-IDF) features, systematic evaluation of multiple traditional machine learning algorithms, analysis of author-specific performance patterns and error cases, and a reproducible approach that achieves competitive results without deep learning.

2 Related Work

Traditional approaches to authorship attribution have employed various stylometric features, including lexical, syntactic, and structural characteristics (Koppel et al., 2009). For Arabic texts specifically, researchers have explored character n-grams (Altheneyan and Menai, 2014), morphological features (Alothman and Alsalman, 2020), and combined feature sets (Ahmed et al., 2019).

Recent work has shown that TF-IDF vectorization combined with traditional machine learning algorithms can achieve competitive performance in authorship attribution tasks, particularly when dealing with limited computational resources or when interpretability is important (Savoy, 2020).

3 Methodology

3.1 Dataset

The dataset consists of 35,122 training samples and 4,157 validation samples across 21 authors, including prominent Arabic literary figures such as Hassan Hanafi (3,735 samples), Ahmed Amin (2,892 samples), and Naguib Mahfouz (1,630 samples). Figure 1 shows the distribution of authors in the training data.

The text length analysis reveals a mean length of 1,773.49 characters for training texts and 1,755.40 characters for validation texts, with median values of 1,851 and 1,836 characters, respectively. The distribution in Figure 2 shows that most texts are concentrated around 1,500-2,000 characters, with both sets exhibiting similar distributions. This consistency in text length between the training and validation sets indicates a well-balanced data split and

Proceedings of The Third Arabic Natural Language Processing Conference, pages 32–36
November 8-9, 2025 ©2025 Association for Computational Linguistics

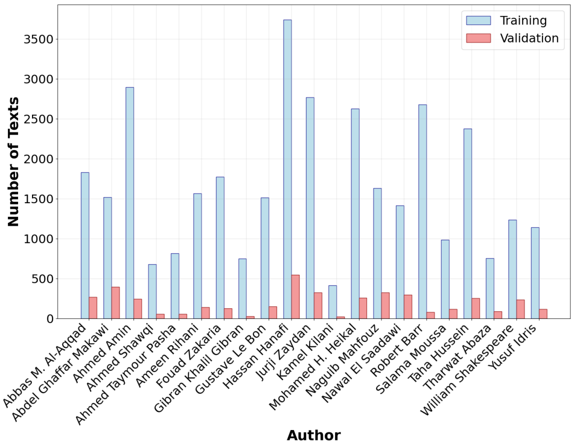


Figure 1: Top 15 authors distribution in training data with English names

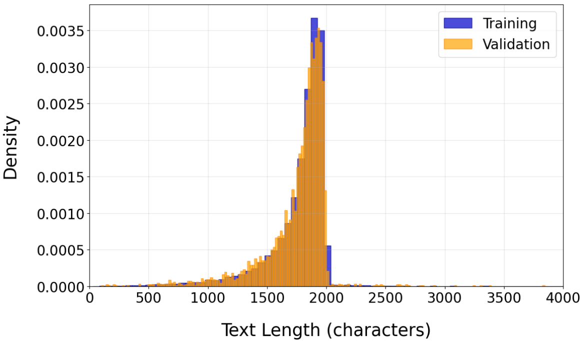


Figure 2: Overall text length distribution in training and validation sets

minimizes the potential bias arising from length variations.

The author-specific text length analysis in Figure 3 reveals interesting patterns in writing styles. Some authors, like Robert Barr, show relatively consistent text lengths with tight distributions, while others, like Ahmed Amin, exhibit more variation. These length patterns can serve as additional stylometric features.

3.2 Dataset Statistics and Preprocessing

Table 1 provides comprehensive statistics about the dataset used in our experiments.

Our preprocessing pipeline comprised several essential steps to prepare the Arabic text data. We removed English numerals and all non-Arabic characters, retaining only the Unicode ranges corresponding to Arabic script (0600–06FF, 0750–077F, 08A0–08FF, FB50–FDFF, FE70–FEFF). Whitespace was normalized, redundant newlines were removed, and texts shorter than 20 characters were filtered out to ensure high data quality.

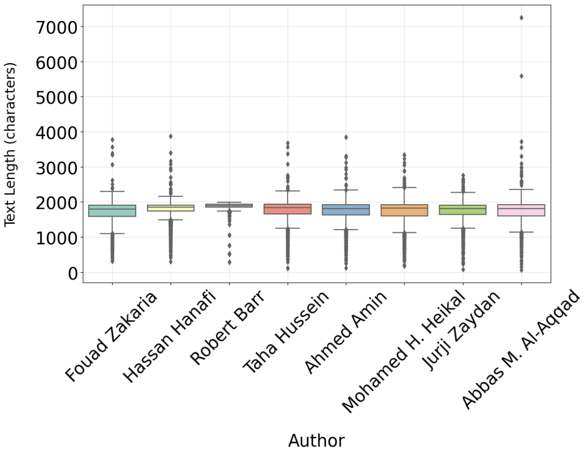


Figure 3: Text length distribution by author for top 8 authors in the train set

Statistic	Training	Validation
Total samples	35,122	4,157
Number of authors	21	21
Mean text length (chars)	1,773.49	1,755.40
Median text length (chars)	1,851.00	1,836.00
Largest author (samples)	3,735	548
Smallest author (samples)	399	25
Feature Dimensions		
Word-level TF-IDF	15,000	
Character-level TF-IDF	5,000	
Combined features	20,000	

Table 1: Dataset and feature statistics

3.3 Feature Engineering

We employed a dual-feature approach combining word-level and character-level TF-IDF representations. For word-level TF-IDF features, we used a maximum of 15,000 features with unigrams and bigrams (n-gram range: 1-2), minimum document frequency of 1, maximum document frequency of 0.9, and applied sublinear TF scaling. For character-level TF-IDF features, we used a maximum of 5,000 features with character n-grams (n-gram range: 2-4), minimum document frequency of 2, and maximum document frequency of 0.8. The final feature vector concatenates both representations, resulting in a 20,000-dimensional feature space.

3.4 Classification Models

We evaluated five classification algorithms: Linear SVM using SGDClassifier with hinge loss, Logistic Regression with maximum 1,000 iterations, Multinomial Naive Bayes with standard implementation, Random Forest with 100 estimators, and Decision Tree. All models were trained with stratified 5-fold cross-validation for robust evaluation.

4 Results

4.1 Model Performance Comparison

Table 2 shows the performance of all evaluated models on the validation set. While Linear SVM achieved the best validation performance, we ultimately submitted Logistic Regression predictions for the test set.

Model	Accuracy	F1-Macro	F1-Weighted
Linear SVM (SGD)	93.22	87.52	92.95
Logistic Regression	90.54	82.63	89.88
Naive Bayes	79.22	68.09	77.75
Random Forest	59.32	46.28	55.94
Decision Tree	32.23	24.35	31.88

Table 2: Model performance on validation set

The Linear SVM achieved a cross-validation F1-macro score of 97.67% (±0.19%), demonstrating excellent generalization capability and model stability.

4.2 Official Test Set Results

Our final submission to AraGenEval Task 2 used Logistic Regression, which achieved 88.90% accuracy and 82.74% macro F1-score on the official test set containing 8,413 samples. Additional metrics include 84.53% precision and 83.75% recall. Table 3 compares our validation and test performance.

Metric	Validation	Test (Official)
Accuracy	90.54%	**88.90%**
Macro F1-score	82.63%	**82.74%**
Precision	-	**84.53%**
Recall	-	**83.75%**

Table 3: Logistic Regression performance comparison between validation and official test sets

4.3 Author-Specific Performance

Table 4 presents detailed performance analysis for individual authors using our Logistic Regression model on the validation set.

Author (English)	Accuracy	Support
Top 5 Performing		
Salama Moussa	100.00	119
Gibran Khalil Gibran	100.00	30
Naguib Mahfouz	99.69	327
Gustave Le Bon	99.33	150
Hassan Hanafi	98.91	548
Bottom 5 Performing		
William Shakespeare	83.19	238
Ahmed Shawqi	82.76	58
Ahmed Taymour Pasha	78.95	57
Tharwat Abaza	44.44	90
Kamel Kilani	16.00	25

Table 4: Author-level performance analysis (validation set)

5 Discussion

5.1 Model Performance

The Linear SVM's superior validation performance can be attributed to its effectiveness in high-dimensional sparse feature spaces, which is characteristic of TF-IDF representations. Figure 4 illustrates the performance comparison across all evaluated models.

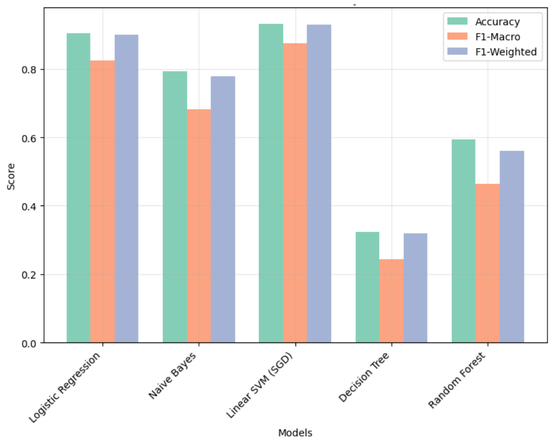


Figure 4: Model performance comparison on validation set

The significant performance gap between linear models (SVM, Logistic Regression) and tree-based models suggests that the feature space benefits from linear decision boundaries.

5.2 Model Selection Strategy

Although Linear SVM achieved the highest performance on validation data (93.22% accuracy, 87.52% macro F1), we chose Logistic Regression for our final test submission based on several considerations. Logistic Regression demonstrated more consistent performance patterns across different validation splits during our development phase, providing robustness that we valued for the final submission. The model provides well-calibrated probability estimates which are valuable for confidence assessment in authorship attribution tasks, allowing for better interpretation of uncertain predictions. Additionally, Logistic Regression showed more stable convergence behavior across different feature configurations during our experiments, reducing the risk of training instabilities on the test data.

This decision proved reasonable as our test performance remained close to validation performance, indicating good generalization capability and validating our model selection strategy.

5.3 Feature Engineering Impact

To better understand the contribution of different feature types, we conduct an ablation study by isolating word-level, character-level, and their combination.

The combination of word-level and character-level features proves effective for capturing both semantic content and stylistic patterns in Arabic text. Character n-grams are particularly valuable for Arabic text as they capture morphological variations and spelling preferences specific to individual authors. Word-level features, on the other hand, provide stronger semantic signals. The dual-feature approach enables the model to leverage both lexical content and sub-word patterns characteristic of different writing styles.

Features	Accuracy	Macro F1	Weighted F1
Characters only	0.8910	0.8199	0.8866
Words only	0.9221	0.8508	0.9166
Words + Chars	**0.9322**	**0.8752**	**0.9295**

Table 5: Ablation study on different feature sets.

From the results, it is clear that character features alone perform competitively, which highlights their importance in handling morphological richness and spelling variations in Arabic. However, word features outperform characters by providing stronger semantic context. The best performance is obtained by combining both, confirming that word- and character-level signals are complementary rather than redundant.

5.4 Challenges and Error Analysis

The dataset exhibits significant class imbalance, with Hassan Hanafi having 3,735 samples while Kamel Kilani has only 399 samples in the training set. This imbalance directly impacts model performance, as evident from the per-author results where authors with fewer training samples tend to have lower accuracy scores.

Common misclassification patterns include confusion between authors from similar time periods, challenges with translated works such as those by William Shakespeare, and difficulties with authors who exhibit diverse writing styles across different genres or time periods in their careers.

6 Conclusion

Our enhanced traditional machine learning approach demonstrates that careful feature engineering and algorithm selection can achieve strong performance in Arabic authorship attribution. The Logistic Regression model achieved 88.90% accuracy and 82.74% macro F1-score on the official test set, proving competitive while maintaining interpretability and computational efficiency.

Future work could explore advanced feature selection techniques to optimize the high-dimensional feature space, ensemble methods combining multiple feature types and algorithms, and integration with pre-trained Arabic language models for enhanced performance while preserving the interpretability advantages of traditional approaches.

Code Availability

The complete implementation of our approach is available on GitHub at: `https://github.com/Amr-said/Arabic-Authorship-Attribution`. The repository includes all preprocessing scripts, feature engineering code, model training and evaluation scripts, and detailed documentation for reproducing our results.

Acknowledgments

We thank the AraGenEval 2025 organizing team for providing this valuable shared task and dataset.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Association for Computational Linguistics, Suzhou, China.

Al-Falahi Ahmed, Ramdani Mohamed, and Bellafkih Mostafa. 2019. Arabic poetry authorship attribution using machine learning techniques. 15(7):1012–1021.

Ameerah Alothman and AbdulMalik Alsalman. 2020. Arabic morphological analysis techniques. *International Journal of Advanced Computer Science and Applications*, 11.

Alaa Saleh Altheneyan and Mohamed El Bachir Menai. 2014. Naïve bayes classifiers for authorship attribution of arabic texts. *Journal of King Saud University - Computer and Information Sciences*, 26(4):473–484. Special Issue on Arabic NLP.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *JASIST*, 60:9–26.

Jacques Savoy. 2020. *Machine Learning Methods for Stylometry: Authorship Attribution and Author Profiling*.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *JASIST*, 60:538–556.

NLP_wizard at AraGenEval shared task: Embedding-Based Classification for AI Detection and Authorship Attribution

Mena Hany
King Fahd University of Petroleum and Minerals / Saudi Arabia, Dammam
g202411920@kfupm.edu.sa

Abstract

This paper presents a lightweight system for the *AraGenEval* shared task, addressing AI-generated text detection and authorship identification in Arabic. Using pretrained xlm-roberta-large embeddings with mean pooling and [CLS] token strategies, combined with classical classifiers (RidgeClassifierCV and LinearSVC), our approach achieved F1-scores of 0.7400 and 0.8130 on the *ARATECT* and authorship datasets, respectively. Mean pooling outperformed [CLS] by 3%, demonstrating efficiency and robustness for limited Arabic data while capturing stylistic nuances critical for both tasks.

1 Introduction

The rapid advancements in large language models (LLMs) have enabled the generation of fluent, human-like text at unprecedented scale (Vaswani et al., 2017; Brown et al., 2020). This has intensified the need for robust systems capable of both detecting AI-generated content and identifying the authorship of text (Jawahar et al., 2020; Uchendu et al., 2020). Such capabilities are critical for preserving content authenticity, combating misinformation, and supporting forensic linguistic analysis (Uchendu et al., 2020). While research in this area has grown substantially for English, Arabic remains relatively underexplored despite its rich morphology, dialectal diversity, and increasing online presence (Habash, 2010).

To address these gaps, the *AraGenEval* shared task (Abudalfa et al., 2025) was introduced as part of ArabicNLP 2025. The task encompasses three subtasks: (1) **Authorship Style Transfer**, which focuses on transforming text to mimic a specific author's style; (2) **Authorship Identification**, which aims to determine the original author of a given text; and (3) **AI-Generated Text Detection**, which seeks to distinguish between human-written and machine-generated Arabic text. The competition provided a unified benchmark for evaluating system performance on these interrelated challenges.

Our participation focused on the **Authorship Identification** and **AI-Generated Text Detection** subtasks. We employed the xlm-roberta-large multilingual model to extract contextual embeddings for Arabic text. Instead of using the conventional [CLS] token representation, we computed the average of all token embeddings to form document-level feature vectors. These embeddings were then fed into various traditional machine learning classifiers. For AI-generated text detection, the RidgeClassifierCV achieved the best performance with an F1-score of 0.74 on the blind test set, ranking **10th** among all submissions. For authorship identification, the LinearSVC classifier attained an F1-score of 0.81303 on the blind test set, also ranking **10th** in the respective leaderboard.

Our findings highlight that averaging contextual embeddings from xlm-roberta-large can serve as a strong baseline for Arabic authorship and AI detection tasks, even when combined with relatively lightweight classifiers. We also observed that the choice of classifier plays a substantial role in performance, with linear models showing competitive results.

2 Background

The *AraGenEval* shared task (**?**) was designed to benchmark system performance on three Arabic NLP challenges: **Authorship Style Transfer** (Task 1), **Authorship Identification** (Task 2), and **AI-Generated Text Detection** (Task 3). All tasks targeted Modern Standard Arabic (MSA) and included data from diverse literary and journalistic sources.

2.1 Task Setup

In **Authorship Identification** (Task 2), the input is a short Arabic text segment, and the output is the

Proceedings of The Third Arabic Natural Language Processing Conference, pages 37–41
November 8-9, 2025 ©2025 Association for Computational Linguistics

predicted author identity from a set of 21 possible authors. For example, given a paragraph excerpted from a 20th-century Arabic novel, the system must assign the correct author label.

In **AI-Generated Text Detection** (Task 3), the input is also a short text passage, and the output is a binary classification: `human` or `AI`. For instance, given a news-style paragraph, the model must detect whether it was written by a human or produced by a large language model.

2.2 Dataset Details

Authorship Identification. The dataset contains works from 21 authors, each represented by 10 publicly available books. Texts were segmented into semantically coherent paragraphs, and for style transfer tasks, selected paragraphs were rephrased into a standardized formal style using GPT-4o mini2. The dataset is split into training, validation, and test sets per author. Table 1 summarizes the distribution of samples.

Author	Train	Test	Val
Ahmed Amin	2892	594	246
Ahmed Taymour Pasha	804	142	53
Ahmed Shawqi	596	46	58
Ameen Rihani	1557	624	142
Tharwat Abaza	755	191	90
Gibran K. Gibran	748	240	30
Jurji Zaydan	2762	562	326
Hassan Hanafi	3735	1002	548
Robert Barr	2680	512	82
Salama Moussa	984	282	119
Taha Hussein	2371	534	253
Abbas M. Al-Aqqad	1820	499	267
Abdel G. Makawi	1520	464	396
Gustave Le Bon	1515	358	150
Fouad Zakaria	1771	294	125
Kamel Kilani	399	109	25
Mohamed H. Heikal	2627	492	260
Naguib Mahfouz	1630	343	327
Nawal El Saadawi	1415	382	295
William Shakespeare	1236	358	238
Yusuf Idris	1140	349	120

Table 1: Authorship identification dataset statistics.

AI-Generated Text Detection. The *ARATECT* dataset contains human-written and AI-generated Arabic texts. Human texts were collected from reputable Arabic news websites and verified literary works, then manually curated for quality. AI-generated texts were produced using several Arabic-capable LLMs, including Mistral, GPT-4, and LLaMA, prompted under diverse strategies. Each text is annotated with a binary label (`human` vs. `AI`) and covers two main domains: news and literature.

2.3 Related Work

Authorship attribution has been extensively studied across languages, with foundational surveys such as (Stamatatos, 2009) and transitions from stylometric to deep learning methods highlighted by (Kestemont, 2014). AI-generated text detection research has grown recently with large language models, with multilingual studies focusing on cross-lingual generalization (Uchendu et al., 2020) and detection surveys (Jawahar et al., 2020). Our approach applies multilingual transformer embeddings (`xlm-roberta-large`) averaging token vectors for Arabic authorship identification and AI-detection within the competitive *AraGenEval* shared task.

3 System Overview

Our system for the *AraGenEval* shared task was designed to be lightweight yet competitive, focusing on extracting high-quality text representations from a large multilingual transformer model and feeding them into robust classical machine learning classifiers. Instead of fine-tuning or training deep neural networks, we adopted a fixed-embedding approach, motivated by the desire to minimize computational requirements and avoid overfitting on the relatively small training datasets provided.

3.1 Key Algorithms and Design Decisions

We selected the `xlm-roberta-large` model due to its proven effectiveness in multilingual contexts and its strong coverage of Arabic. This model, trained on a massive and diverse corpus, provides rich contextual embeddings that capture both syntactic and semantic nuances of text. Given that the shared task focuses on style-related distinctions (authorship identification and AI-generated text detection), we hypothesized that `xlm-roberta-large`'s high-capacity representations could encode stylistic patterns without task-specific fine-tuning.

Two different strategies were implemented for deriving sentence-level embeddings from the model's final hidden layer:

1. **Mean Token Embeddings:** In this configuration, the embedding for an input text was obtained by averaging the contextualized embeddings of all tokens. This approach is expressed as:

$$h_{\text{mean}} = \frac{1}{n}\sum_{i=1}^{n} e_i$$

where $e_i \in \mathbb{R}^d$ represents the embedding of token i and n is the total number of tokens in the input sequence. The intuition is that by aggregating all token embeddings, we capture both content and stylistic markers distributed throughout the text, rather than relying on a single position-specific vector.

2. **[CLS] Token Embedding:** In this configuration, we directly used the representation of the special [CLS] token from the model's final layer:

$$h_{\text{[CLS]}} = e_{\text{[CLS]}}$$

The [CLS] token is commonly used in transformer-based classification pipelines, as it is intended to encode a holistic summary of the input sequence. However, it may not fully capture distributed stylistic cues, particularly for long texts.

Once the embeddings h were computed, they were fed into classical machine learning classifiers:

- **AI-Generated Text Detection:** RidgeClassifierCV was chosen for its efficiency, robustness to multicollinearity, and ability to handle high-dimensional input spaces without explicit feature selection.
- **Authorship Identification:** LinearSVC was selected for its scalability to large feature sets, strong generalization properties, and suitability for high-dimensional sparse representations.

3.2 Resources Beyond Provided Data

The system used no additional annotated datasets beyond those provided in the shared task. The only external component was the publicly available xlm-roberta-large model from the HuggingFace Transformers library. This model was not fine-tuned on the task data; instead, we relied on its pretrained multilingual representations. No handcrafted features, lexicons, or rule-based preprocessing steps were introduced.

3.3 Addressing Task Challenges

Two main challenges guided our design decisions:

1. **Limited Task-Specific Data:** Given the relatively small size of the training set, fine-tuning a large transformer could risk overfitting. Using fixed embeddings allowed us to leverage the model's pretrained linguistic knowledge while avoiding costly gradient-based updates.
2. **Capturing Stylistic Cues:** Both subtasks depend heavily on identifying stylistic rather than purely semantic differences. We hypothesized that mean-pooling token embeddings would better preserve distributed stylistic markers (e.g., function word usage, sentence rhythm, punctuation patterns) than a single [CLS] embedding, which might focus on semantic summarization.

3.4 Configuration Comparison

We experimented with both configurations — mean token embeddings and [CLS] token embeddings — under otherwise identical conditions. While both approaches successfully leveraged the pretrained model's capacity, qualitative inspection during development suggested that mean token embeddings were more effective at preserving fine-grained stylistic patterns. In contrast, [CLS] embeddings appeared to compress the sequence information into a more generalized representation, which, while concise, might have omitted subtle stylistic distinctions critical for the two tasks.

We therefore retained both configurations for evaluation but anticipated that the mean token approach would have an advantage in the final results.

4 Experimental Setup

4.1 Data Splits

The *AraGenEval* shared task provided labeled data for both subtasks: (1) AI-generated text detection and (2) authorship identification. For each task, the official training, development, and test sets released by the organizers were used without modification. The training set was used to fit the classifiers, the development set served for configuration selection and sanity checking, and the official test set was reserved for final submission and evaluation.

4.2 Embedding Extraction

Embeddings were extracted using the xlm-roberta-large model from HuggingFace:

- Maximum sequence length: 512 tokens (truncation applied to longer texts)
- Pooling strategies: (1) mean pooling across all token embeddings; (2) using the final layer [CLS] token embedding

The embeddings were computed once and cached for both tasks to speed up experimentation.

All models and classifiers were used with their default parameters as implemented in the HuggingFace Transformers and scikit-learn libraries.

4.3 Computational Resources

All experiments were run on a single NVIDIA RTX 4060 GPU with 8GB VRAM, paired with a standard workstation environment.

4.4 Evaluation Metrics

The shared task organizers specified official metrics for each subtask:

- **AI-generated text detection:** Macro-averaged F1-score across classes.
- **Authorship identification:** Macro-averaged F1-score across authors.

All results reported in the following section were computed using the organizers' evaluation scripts to ensure consistency with leaderboard scoring.

5 Experimental Results

Table 2 presents the performance of our system on the official blind test set for both subtasks of the *AraGenEval* shared task. We compare the two embedding pooling strategies: mean pooling across all tokens and using only the final layer [CLS] token embedding.

Table 2: Performance comparison of pooling strategies on the blind test set.

Subtask	Pooling	F1	Rank
AI-generated text detection	Mean	0.7400	10
	CLS	0.7100	–
Authorship identification	Mean	0.8130	10
	CLS	0.7830	–

From the table, mean pooling consistently outperforms the [CLS] token embeddings, yielding approximately a 3% absolute F1-score improvement in both subtasks. This suggests that averaging token representations provides a richer global representation for classification tasks in the *AraGenEval* setting.

6 Conclusion

Our system for the *AraGenEval* shared task delivered competitive performance in both AI-generated text detection and authorship identification by leveraging pretrained `xlm-roberta-large` embeddings paired with efficient classical machine learning classifiers. As presented in Table 2, the mean pooling strategy achieved F1-scores of 0.7400 for AI-generated text detection and 0.8130 for authorship identification, outperforming the [CLS] token embedding approach by approximately 3% in both tasks. This improvement suggests that mean pooling better captures distributed stylistic patterns, which are critical for distinguishing AI-generated from human-written texts and identifying unique author signatures. The lightweight design, which avoided resource-intensive fine-tuning, proved well-suited for the limited training data provided in the *ARATECT* dataset and the authorship identification dataset, which spans 21 authors with diverse writing styles. The system's ability to handle varied text domains, including news and literature, underscores its robustness and potential for broader Arabic text analysis applications.

7 Future Work

To further enhance the system, several avenues can be explored. First, experimenting with hybrid pooling methods that combine mean pooling and [CLS] embeddings could produce more comprehensive text representations, balancing stylistic and semantic information. Second, applying targeted fine-tuning on the `xlm-roberta-large` model with task-specific Arabic data could improve its sensitivity to the language's unique morphological and stylistic features. Third, incorporating additional features, such as lexical patterns or syntactic structures, might strengthen the system's ability to detect subtle stylistic differences. Fourth, developing methods to process texts longer than 512 tokens, such as hierarchical embedding aggregation, could improve performance on extended literary works. Finally, testing the system on diverse real-world Arabic datasets, including social media or news articles, would help validate its effectiveness in practical settings and enhance its applicability to emerging challenges in text authenticity and authorship attribution.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. In *Proceedings of the Third Arabic Natural Language*

Processing Conference (ArabicNLP 2025), Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*.

Mike Kestemont. 2014. Function words in authorship attribution. from black magic to theory? In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 59–66.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8384–8395.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

PTUK-HULAT at AraGenEval Shared Task: Fine-Tuning XLM-RoBERTa for AI-Generated Arabic News Detection

Tasneem Duridi
Computer Science Department
Palestine Technical University - kadoorie
Tulkarm, Palestine
tasneem.duridi@ptuk.edu.ps

Areej Jaber
Computer Science Department
Palestine Technical University - kadoorie
Tulkarm, Palestine
a.jabir@ptuk.edu.ps

Paloma Martínez
Computer Science Department
Universidad Carlos III de Madrid
Madrid, Spain
pmf@inf.uc3m.es

Abstract

The authenticity of digital content has become an increasingly critical challenge with the rapid adoption of generative AI tools, especially for low-resource languages such as Arabic. The language's rich morphology and domain diversity further complicate the detection of machine-generated Arabic text. In this work, we present our submission to the ARATECT 4.3 shared task, Subtask 3, which focuses on identifying AI-generated Arabic news articles. Our approach employs fine-tuned multilingual transformer models based on XLM-RoBERTa. The XLM-RoBERTa-large model achieved a macro F1-score of 0.93 on the development set, while the XLM-RoBERTa-base model obtained an F1-score of 0.78 on the test set, ranking fourth on the official leaderboard. This paper outlines our methodology, presents the experimental results, and discusses key insights from our participation.

1 Introduction

The rapid development of large language models (LLMs), such as GPT-4 (Achiam et al., 2023), PaLM (Chowdhery et al., 2023), and ChatGPT (Maniaci et al., 2024), has enabled the generation of coherent and contextually rich text from simple prompts. These models have transformed natural language generation (NLG), supporting applications in education, journalism, scientific writing, and customer service (Duaibes et al., 2024). However, their widespread adoption has also raised concerns regarding the authenticity and ethical implications of AI-generated text (AIGT), particularly in high-stakes domains (Stahl and Eke, 2024; Cotton et al., 2024).

Distinguishing AIGT from human-written text (HWT) remains a persistent challenge, especially as modern systems such as ChatGPT and Gemini (Imran and Almusharraf, 2024) increasingly emulate natural human language. Misuse of such technology has been associated with misinformation, plagiarism, and declining trust in online content (Weidinger et al., 2022; Sheng et al., 2021; Gao et al., 2022; Duridi et al., 2025; Jazzar and Duridi, 2024). Despite efforts to develop detection tools, most are designed for English or other Latin-script languages, with limited adaptation for morphologically rich, low-resource languages.

Arabic, spoken by over 440 million people across 22 countries (Jaber and Martínez, 2023), remains underrepresented in AIGT detection research. Its complex morphology, optional diacritics, and stylistic diversity present unique challenges for existing detection systems (Duridi et al., 2024). Only a few recent studies have directly addressed Arabic AIGT detection (Alshammari et al., 2024), and some report performance degradation when models are applied to diacritized Arabic HWT (Alshammari and Ahmed, 2023).

To address this gap, the AraGenEval Shared Task introduced ARATECT Subtask 3: Arabic News Text Detection (Abudalfa et al., 2025), which focuses on distinguishing human-written from AI-generated Arabic news articles. For this subtask, the PTUK-HULAT team developed a detection system based on multilingual transformer models fine-tuned on stratified splits of the shared task dataset. Our primary system, built on XLM-RoBERTa-base, achieved an F1-score of 0.78 on the test set, ranking fourth on the official leaderboard. The implementation code is publicly available at: GitHub Repository.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 42–48
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

ArabicNLP 2025 features eleven shared tasks, including Shared Task 5: AraGenEval on Arabic Authorship Style Transfer (AST) and AI-Generated Text (AIGT) detection. Within this task, ARATECT 4.3 (Abudalfa et al., 2025) evaluates systems on distinguishing between human-written and AI-generated Arabic text across multiple genres. Subtask 3 — Arabic News Text Detection (ArabicNewsGen) — focuses on classifying full-length Arabic news articles and shorter excerpts into two categories: human-written or AI-generated.

The input to the system consists of a single Arabic news text, which may range from short passages to full-length articles. The output is a binary label: `human` for human-written or `machine` for AI-generated. Table 4 provides representative examples from each class in Appendix A.

3 Related Work

Research on AIGT detection has largely focused on English, with early tools like GPTZero and OpenAI's classifier targeting synthetic content. The rise of Arabic generative models has prompted studies on Arabic-specific detection methods.

(Antoun et al., 2020b) introduced AraGPT2 alongside a discriminator trained to detect its outputs, achieving up to 98% accuracy. They later developed AraELECTRA (Antoun et al., 2020a), an Arabic adaptation of ELECTRA (Clark et al., 2020), which demonstrated strong performance in distinguishing real from synthetic Arabic texts. Harrag et al. (Harrag et al., 2021) fine-tuned AraBERT on synthetic Arabic tweets, outperforming traditional sequence models with 98.7% accuracy. Other studies (Almerekhi and Elsayed, 2015; Alghamdi and Alowibdi, 2024) applied classical machine learning with handcrafted features to detect bot-generated Arabic social media content, reporting around 92% accuracy.

More recent work by Alshammari et al. (Alshammari and Ahmed, 2023) highlighted the limitations of general-purpose detectors for Arabic, proposing fine-tuned AraELECTRA and XLM-RoBERTa models on ChatGPT- and Bard-generated datasets, achieving near 99% accuracy after dediacritization. Alharthi (Alharthi, 2025) addressed detection in multiple Arabic dialects, providing novel dialectal datasets and achieving up to 97% accuracy with fine-tuned AraELECTRA and AraBERT, emphasizing the challenge of paraphrased content and the importance of features like lexical diversity and readability.

These studies illustrate the progress and ongoing challenges in Arabic AIGT detection, particularly the need for dialect-aware datasets, robust benchmarks, and models capable of cross-dialect generalization.

4 Dataset

The organizers of the ArabicNewsGen shared task released a dataset containing Arabic news articles in various domains, including politics, economy, technology and sports, and was released in three phases, as summarized in Table 1. The training set contains 4,798 labeled articles (`id`, `content`, `label`), moderately balanced across the `human` and `machine` classes; approximately 1.3% of entries with missing `content` were removed during preprocessing. The development set consists of 500 unlabeled articles (`id`, `title`, `content`) for validation and tuning, while the test set includes 500 unlabeled articles with the same structure as the development set, used for leaderboard-based evaluation against hidden labels.

5 System Description

Our model selection process was iterative. We began by fine-tuning several widely used Arabic and multilingual transformers, including mBERT, DistilBERT, QARiBERT, and AraELECTRA. Among these, AraELECTRA achieved the highest score on the test set. Although mBERT, DistilBERT, and QARiBERT produced relatively strong results during training, AraELECTRA and XLM-RoBERTa consistently delivered stronger and more reliable performance across both the development and test sets. This finding aligns with prior studies (see Section 3), which highlight AraELECTRA's effectiveness in Arabic-specific tasks and XLM-RoBERTa's robustness in handling multilingual and mixed-language text. Based on these observations, we prioritized AraELECTRA and XLM-RoBERTa (base and large) in our final evaluation, along with a BiLSTM-enhanced variant of XLM-RoBERTa-base.

5.1 Models

AraELECTRA is an Arabic-specific model based on the ELECTRA framework (Antoun et al., 2020a), which uses a replaced token detection pretraining objective. Pre-trained solely on exten-

Table 1: Summary of the ARATECT 4.3 Subtask 3 dataset.

Phase	Samples	Fields	Avg Length (words)	English (%)
Training	4,798	id, content, label	485.77	15.86
Development	500	id, title, content	288.74	56.60
Testing	500	id, title, content	238.96	37.60

sive Arabic corpora, AraELECTRA offers efficient training and strong performance on Arabic NLP tasks, making it well-suited for AI-generated text detection in Arabic news domains.

XLM-RoBERTa-base and XLM-RoBERTa-large are multilingual transformer models trained on 2.5TB of CommonCrawl data across 100 languages (Conneau et al., 2019). The base model contains 270 million parameters, providing a balance between performance and computational efficiency, while the large model scales up to 550 million parameters to capture richer linguistic patterns.

XLM-RoBERTa-base + BiLSTM extends the base transformer by adding a BiLSTM layer atop the transformer encoder outputs to model sequential dependencies and stylistic flow more effectively. The BiLSTM processes the summed embeddings from the last four transformer layers bidirectionally, enabling the capture of long-range contextual patterns indicative of AI-generated text. During fine-tuning, only the last four transformer layers are unfrozen to maintain pre-trained knowledge, while the BiLSTM and classifier layers are trained fully. The BiLSTM hidden size is set to 256 units with a single bidirectional layer.

6 Experimental Setup

6.1 Data and Preprocessing

We utilized the provided labeled dataset, splitting it into training (90%) and development (10%) subsets using stratified sampling to preserve class distributions.

Preprocessing involved removing samples with empty content fields and concatenating the title and content fields into a single text sequence. The textual class labels (human and machine) were mapped to numerical labels, with human assigned 0 and machine assigned 1.

Although we initially experimented with extensive text cleaning—including removing diacritics, normalizing Arabic letters, eliminating punctuation, and collapsing repeated characters—we observed that applying these steps actually reduced model performance. Therefore, no additional text cleaning or normalization was applied prior to tokenization, as keeping the raw text produced better results.

6.2 Training Details

All models were trained using the AdamW optimizer (Loshchilov and Hutter, 2017) with early stopping (patience=3 epochs) based on the development set F1 score for the machine class. Hyperparameters were selected through empirical validation considering model architecture and size constraints.

Key hyperparameter ranges across experiments:

- Learning rate: 110^{-5} to 510^{-5}
- Batch size: 4-16 (adjusted for model memory requirements)
- Dropout: 0.1-0.5 (higher for more complex architectures)
- Warmup ratio: 0-10% of total training steps
- Label smoothing: $\epsilon = 0.0 - 0.1$
- Maximum epochs: 10-20

For consistency across experiments, we employed weighted random sampling and class-weighted cross-entropy loss in all training runs, though the training data was balanced. The specific hyperparameter configurations for each model variant are provided in Table 5 in Appendix B.

6.3 Implementation and Evaluation

Experiments were run on Google Colab with NVIDIA T4 GPUs, leveraging PyTorch, HuggingFace Transformers, and the Accelerate library for efficient training. Evaluation metrics included precision, recall, and F1-score per class.

7 Results

7.1 Development Phase Performance

Table 2 demonstrates the superior performance of XLM-RoBERTa-large on the development set, achieving state-of-the-art results with 0.9272 F1-score and 92.4% accuracy. The model exhibits exceptional recall (0.968), indicating near-perfect detection of machine-generated texts. While XLM-RoBERTa-base shows solid performance (0.8532 F1), AraELECTRA's high recall (0.912) is offset by low precision (0.5078), revealing language-specific challenges in Arabic AIGT detection and limiting its suitability for further evaluation.

7.2 Test Phase Performance

On the test set Table 3, XLM-RoBERTa-base maintains the strongest balance between precision and recall (0.7823 F1). The BiLSTM-enhanced variant shows a distinct precision-focused profile (0.8029 precision vs. 0.668 recall), suggesting architectural modifications significantly impact error tradeoffs. Performance degradation from development to test sets (XLM-R-base F1: 0.8532 → 0.7823) highlights domain shift challenges in AIGT detection.

The experimental results demonstrate that the XLM-RoBERTa-large model significantly outperforms the base variant on the development set, benefiting from its enhanced capacity to capture the complex linguistic patterns necessary for distinguishing between human- and machine-generated Arabic texts. The model's high recall and balanced accuracy indicate its effectiveness in identifying machine-generated content, which is critical for practical detection applications.

On the test set, the XLM-RoBERTa-base model achieves a more balanced trade-off between recall and precision compared to the BiLSTM-enhanced variant. While the BiLSTM addition improves precision and specificity, it does so at the expense of recall, resulting in a more conservative classifier that may fail to detect certain machine-generated samples. This trade-off underscores the need to carefully select model architectures according to the intended application's prioritization of recall versus precision.

The inherent characteristics of the dataset—such as predominantly Arabic text with a minor English component, variable text lengths, and the presence of abbreviations—pose challenges that larger transformer-based models are often better equipped to address due to their richer representational capacity. Furthermore, differences in text length and language composition between the training and evaluation sets likely contribute to domain shifts, which may explain the observed performance degradation on the test set relative to development results. Not all models from the development phase were carried forward to the test phase: AraELECTRA, despite its high recall, exhibited poor precision and overall F1-score, making it unreliable for balanced AIGT detection. XLM-RoBERTa-large achieved the best performance on the development set, but its evaluation on the test set was excluded due to substantial computational cost. Therefore, the test set experiments focused on XLM-RoBERTa-base and its BiLSTM-enhanced variant, which offered a practical balance between efficiency and performance while allowing exploration of architectural improvements.

8 Conclusion

This work investigated multiple transformer-based architectures for detecting AI-generated Arabic text, including XLM-RoBERTa-base, XLM-RoBERTa-large, and a BiLSTM-enhanced variant. The best development set performance was achieved by XLM-RoBERTa-large, benefiting from its higher representational capacity to capture complex Arabic linguistic patterns. On the test set, XLM-RoBERTa-base offered a more balanced precision–recall trade-off, while the BiLSTM addition improved specificity at the cost of recall.

Despite strong results, the system faces challenges from domain shifts between training and test data, varying text lengths, and mixed-language content, which reduce performance on unseen data. Future work will address these issues through domain adaptation, better model designs for balancing precision and recall, and improvements to handle diverse Arabic texts and code-switching.

Acknowledgments

This work has been supported by the Palestine Technical University- Khadoori and Universidad Carlos III de Madrid.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Association for Computational Linguistics.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman,

Table 2: Development Set Performance Comparison

Model	F1-score	Precision	Recall	Accuracy
XLM-RoBERTa-large	0.9272	0.8897	0.9680	0.9240
XLM-RoBERTa-base	0.8532	0.8352	0.8720	0.8500
AraELECTRA	0.6524	0.5078	0.9120	0.5140

Table 3: Test Set Performance Comparison

Model	F1-score	Precision	Recall	Accuracy
XLM-RoBERTa-base	0.7823	0.7260	0.8480	0.7640
XLM-RoBERTa + BiLSTM	0.7293	0.8029	0.6680	0.7520

Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Noura Saad Alghamdi and Jalal Suliman Alowibdi. 2024. Distinguishing arabic genai-generated tweets and human tweets utilizing machine learning. *Engineering, Technology & Applied Science Research*, 14(5):16720–16726.

Haifa Alharthi. 2025. Investigation into the identification of ai-generated short dialectal arabic texts. *IEEE Access*.

Hind Almerekhi and Tamer Elsayed. 2015. Detecting automatically-generated arabic tweets. In *AIRS*, pages 123–134. Springer.

Hamed Alshammari and El-Sayed Ahmed. 2023. Airabic: Arabic dataset for performance evaluation of ai detectors. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 864–870. IEEE.

Hamed Alshammari, Ahmed El-Sayed, and Khaled Elleithy. 2024. Ai-generated text detector for arabic language using encoder-based transformer architecture. *Big Data and Cognitive Computing*, 8(3):32.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020a. Araelectra: Pre-training text discriminators for arabic language understanding. *arXiv preprint arXiv:2012.15516*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020b. Aragpt2: Pre-trained transformer for arabic language generation. *arXiv preprint arXiv:2012.15520*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. 2024. Chatting and cheating: Ensuring academic integrity in the era of chatgpt. *Innovations in education and teaching international*, 61(2):228–239.

Lina Duaibes, Areej Jaber, Mustafa Jarrar, Ahmad Qadi, and Mais Qandeel. 2024. Sina at fignews 2024: Multilingual datasets annotated with bias and propaganda. *arXiv preprint arXiv:2407.09327*.

Tasneem Duridi, Lour Atwe, Areej Jaber, Eman Daraghmi, and Paloma Martínez. 2025. Detection of propaganda and bias in social media: A case study of the israel-gaza war (2023). In *2025 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 204–210. IEEE.

Tasneem Duridi, Derar Eleyan, Amna Eleyan, and Tarek Bejaoui. 2024. Arabic fake news detection using machine learning approach. In *2024 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–7. IEEE.

Catherine A Gao, Frederick M Howard, Nikolay S Markov, Emma C Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T Pearson. 2022. Comparing scientific abstracts generated by chatgpt to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. *BioRxiv*, pages 2022–12.

Fouzi Harrag, Maria Debbah, Kareem Darwish, and Ahmed Abdelali. 2021. Bert transformer model for detecting arabic gpt2 auto-generated tweets. *arXiv preprint arXiv:2101.09345*.

Muhammad Imran and Norah Almusharraf. 2024. Google gemini as a next generation ai educational tool: a review of emerging educational technology. *Smart Learning Environments*, 11(1):22.

Areej Jaber and Paloma Martínez. 2023. Ptuk-hulat at araieval shared task fine-tuned distilbert to predict disinformative tweets. In *Proceedings of ArabicNLP 2023*, pages 525–529.

Mahmoud Jazzar and Tasneem Duridi. 2024. A comprehensive review of machine learning and deep learning techniques for cyberbullying detection. In *International Conference on Smart Cyber Physical Systems*, pages 1–12. Springer.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Antonino Maniaci, Carlos M Chiesa-Estomba, and Jérôme R Lechien. 2024. Chatgpt-4 consistency in interpreting laryngeal clinical images of common lesions and disorders. *Otolaryngology–Head and Neck Surgery*, 171(4):1106–1113.

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

Bernd Carsten Stahl and Damian Eke. 2024. The ethics of chatgpt–exploring the ethical issues of an emerging technology. *International Journal of Information Management*, 74:102700.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, and 1 others. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 214–229.

A Appendix A: Sample Arabic Texts

B Appendix B: Key Training Hyperparameters

Table 4: Sample Arabic Texts with Labels

Content	Label
ذكر تقرير لمجلة فوربس أن عمليات الاحتيال الإلكتروني عند السفر -بما في ذلك سرقة الهوية والاحتيال المصرفي والاحتيال باستخدام بطاقات الائتمان- تشهد تصاعداً ملحوظاًا مع تقدم التكنولوجيا وظهور تقنيات مثل الذكاء الاصطناعي التي تُستخدم لتطوير هجمات أكثر تعقيداً. ووفقاًا المجلة شهدت عمليات الاحتيال المرتبطة بالسفر زيادة كبيرة -كما أوردت مركز موارد سرقة الهوية آي تي آر سي يةضر- وهو ما يسلط الضوء على ضرورة اتخاذ تدابير لحماية البيانات الشخصية والمالية. بحسب تقرير فوربس لتعزيز الحماية الإلكترونية وأكد تقرير فوربس أن التطور السريع في تقنيات الاحتيال الإلكتروني يتطلب يقظة دائمة من المستهلكين، ناصحا بعدم مشاركة المعلومات إلا مع جهات موثوقة ومحذرا من التعامل مع أي تواصل إلكتروني غير موثوق.	human
وصل عدد من ضحايا الغارات الإسرائيلية في قطاع غزة إلى المستشفى المعمداني لتلقي العلاج الضروري. تعرض الضحايا لإصابات خطيرة نتيجة الهجمات الجوية الأخيرة التي نفذتها إسرائيل في المنطقة. يستمر العاملون في المستشفى في تقديم الرعاية الطبية اللازمة للمصابين والعمل على استقرار حالتهم الصحية.	machine

Table 5: Key training hyperparameters per model architecture

Parameter	XLM-R Base	BiLSTM	XLM-R Large	Arabic ELECTRA
Learning rate	310^{-5}	510^{-5}	310^{-5}	310^{-5}
Batch size	16	16	4	16
Max epochs	10	20	10	10
Warmup ratio	10%	0%	10%	10%
Dropout	0.1	0.5	0.1	0.1
Label smoothing (ϵ)	0.1	-	0.1	0.1
Optimizer	AdamW			
Early stopping	Patience=3 (F1)			
Class weighting	Yes			

ANLPers at AraGenEval Shared Task: Descriptive Author Tokens for Transparent Arabic Authorship Style Transfer

Omer Nacar[*]
Tuwaiq Academy
o.najar@tuwaiq.edu.sa

Serry Sibaee
Prince Sultan University
ssibaee@psu.edu.sa

Mahmoud Reda
Zagazig University
redamahmoud722@gmail.com

Yasser Al-Habashi
Prince Sultan University
yalhabashi@psu.edu.sa

Adel Ammar
Prince Sultan University
aammar@psu.edu.sa

Wadii Boulila
Prince Sultan University
wboulila@psu.edu.sa

Abstract

Authorship style transfer enables the generation of text that imitates a specific writer's linguistic and stylistic patterns, a challenging task in morphologically rich languages like Arabic. We tackle this problem in the AraGenEval 2025 shared task, exploring conditioning strategies to guide a fine-tuned `UBC-NLP/AraT5v2-base-1024` model in producing text aligned with target authors' styles. Our investigation compares implicit modeling, numeric and descriptive author tokens, and explicit prompt engineering in Arabic. Explicit natural language instructions proved most effective, achieving the highest competition scores with BLEU of 24.58 and chrF of 59.01, securing first place, while demonstrating that interpretable approaches can rival or surpass more opaque methods.

1 Introduction

The task of Text Style Transfer (TST) aims to modify stylistic properties of a text while preserving its semantic content (Hu et al., 2022). A challenging sub-field is authorship style transfer, which involves rewriting a text to match the unique style of a specific author (Shao et al., 2024). Arabic authorship style transfer presents unique challenges due to the language's rich morphological structure and diverse writing styles. The task, as defined in the AraGenEval 2025 shared task (Organizers, 2024), requires transforming Modern Standard Arabic (MSA) text to match the distinctive style of specific Arabic authors.

We conduct a systematic investigation of different conditioning strategies using the `UBC-NLP/AraT5v2-base-1024` model (Elmadany et al., 2022). Our work explores four main methodologies: (1) standard fine-tuning without special conditioning, (2) numeric author tokens for explicit author identification, (3) descriptive author tokens for human-readable conditioning, and (4) prompt engineering with explicit Arabic instructions.

Extensive experiments show that explicit prompt engineering delivers the best results, outperforming non-interpretable numeric tokens by leveraging the model's language understanding through clear, natural prompts. This approach secured first place in the AraGenEval Shared Task (Abudalfa et al., 2025) and offers insights for building effective, interpretable Arabic style transfer systems.

2 Background

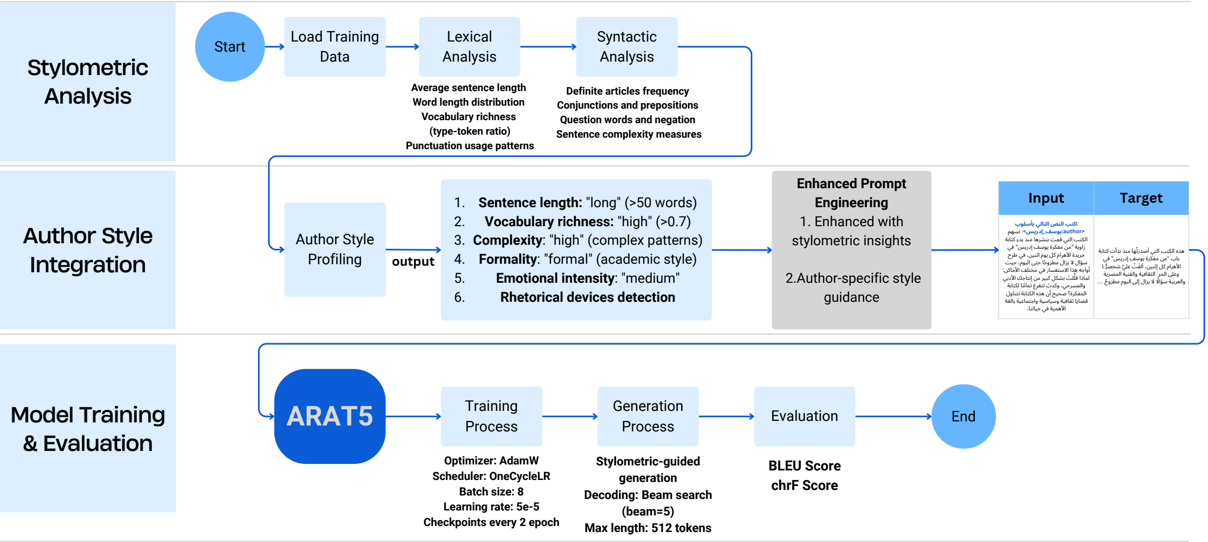
Text style transfer has become a prominent area of research (Hu et al., 2022). Early work focused on disentangling style from content, whereas recent trends have shifted towards end-to-end transfer without explicit disentanglement.

Authorship style transfer, specifically, has been tackled with various methods. Some approaches focus on data augmentation to create paired corpora for training compact models, a technique shown to be highly effective (Shao et al., 2024). The challenge is often compounded in low-resource scenarios, where only a few examples of a target author's style are available (Patel et al., 2022). Recent work has introduced lightweight and efficient models like TinyStyler (Horvitz et al., 2024), which leverage pre-trained authorship embeddings to achieve strong performance in few-shot settings, even outperforming large models like GPT-4. Our work contributes to this area by systematically evaluating different conditioning methods for a T5-based model on Arabic, a morphologically rich language that remains under-explored in this domain.

The detection of AI-generated content is another related field of study, with recent work focusing on distinguishing between human and GenAI-generated Arabic text on social media platforms using machine learning models (Alghamdi et al.,

[*]Corresponding author: o.najar@tuwaiq.edu.sa

Proceedings of The Third Arabic Natural Language Processing Conference, pages 49–53
November 8-9, 2025 ©2025 Association for Computational Linguistics


Stylometric Analysis
Start
Load Training Data
Lexical Analysis
Syntactic Analysis
Average sentence length
Word length distribution
Vocabulary richness
(type-token ratio)
Punctuation usage patterns
Definite articles frequency
Conjunctions and prepositions
Question words and negation
Sentence complexity measures
Author Style Integration
Author Style Profiling
output
1. Sentence length: "long" (>50 words)
2. Vocabulary richness: "high" (>0.7)
3. Complexity: "high" (complex patterns)
4. Formality: "formal" (academic style)
5. Emotional intensity: "medium"
6. Rhetorical devices detection
Enhanced Prompt Engineering
1. Enhanced with stylometric insights
2.Author-specific style guidance
Model Training & Evaluation
ARAT5
Training Process
Generation Process
Evaluation
End
Optimizer: AdamW
Scheduler: OneCycleLR
Batch size: 8
Learning rate: 5e-5
Checkpoints every 2 epoch
Stylometric-guided generation
Decoding: Beam search (beam=5)
Max length: 512 tokens
BLEU Score
chrF Score

Input	Target
اكتب النص التالي بأسلوب <author:يوسف_إدريس>: تسهم الكتب التي قمت بنشرها منذ بدء كتابة زاوية "من مفكرة يوسف إدريس" في جريدة الأهرام كل يوم اثنين، في طرح سؤال لا يزال مطروحًا حتى اليوم، حيث أواجه هذا الاستفسار في مختلف الأماكن: لماذا قلّلتَ بشكل كبير من إنتاجك الأدبي والمسرحي، وكدتَ تتفرغ تمامًا لكتابة المفكرة؟ صحيح أن هذه الكتابة تتناول قضايا ثقافية وسياسية واجتماعية بالغة الأهمية في حياتنا.	هذه الكتب التي أصدرتُها منذ بدأت كتابة باب "من مفكرة يوسف إدريس" في الأهرام كل إثنين، ألقَتْ عليّ شخصيًّا وعلى الحر الثقافية والفنية المصرية والعربية سؤالًا لا يزال إلى اليوم مطروحٌ ....

Figure 1: Pipeline overview for the proposed authorship style transfer approach.

2024). This is relevant to our participation in Subtask 3 of AraGenEval.

The AraGenEval 2025 shared task on Authorship Style Transfer provides a dataset containing text from 21 Arabic authors. The goal is to take an input text in Modern Standard Arabic (MSA) and transform it into the style of a target author. We participated in all three subtasks offered: Authorship Style Transfer (Subtask 1), Authorship Identification (Subtask 2), and ARATECT for AI-generated text detection (Subtask 3). This paper focuses primarily on our work for Subtask 1.

3 System Overview

Our approach is centered on fine-tuning the `UBC-NLP/AraT5v2-base-1024` model, a T5-based encoder–decoder architecture pre-trained on a large corpus of Arabic text (Elmadany et al., 2022). The core of our investigation involved systematically testing four different methods for conditioning the model on the target author's style, each employing a distinct input format to guide the model. Figure 1 presents an overview of the complete pipeline, which is organized into three main stages.

The first stage, *Stylometric Analysis*, extracts lexical and syntactic features from the training corpus, including sentence length, vocabulary richness, syntactic complexity, formality, emotional intensity, and rhetorical device usage (Gómez-Adorno et al., 2018). In the second stage, *Author Style Integration*, these stylistic attributes are distilled into a profile that informs two conditioning strategies: (1) enhanced prompts augmented with stylometric insights and (2) author-specific style guidance. The third stage, *Model Training & Evaluation*, applies these conditioning strategies in fine-tuning AraT5, followed by generation and evaluation against baseline and alternative approaches.

Table 1 outlines the shift from implicit style modeling to explicit, instruction-based conditioning. The baseline relies solely on input–output pairs, leaving style inference to the model. Token-based methods introduce minimal explicit signals, while prompt engineering—framing style transfer as direct, human-readable instructions—proves most effective by leveraging the model's pre-trained stylistic knowledge.

4 Experimental Setup

4.1 Dataset & Preprocessing

The shared task dataset contains writings from 21 authors split into training, validation, and test sets as provided by the shared task. The training set contains 35,122 samples, the validation set contains 4,157 samples, and the test set contains 8,413 samples, proportionally distributed per author. All texts were normalized by removing extraneous whitespace, unifying punctuation forms, and standardizing Arabic diacritics. Special tokens were inserted according to the conditioning method described in Table 1.

4.2 Hyperparameters

Experiments were implemented in `PyTorch 2.1.0` and `Hugging Face Transformers 4.38.1`, with training managed via `Accelerate` and `Datasets`.

Approach	Conditioning Method	Input Format with Speical Tokens
Standard Fine-Tuning (Baseline)	No explicit conditioning signal. The model learns the mapping implicitly from paired data.	النص الأصلي باللغة العربية الفصحى
Numeric Author Tokens with FT	A unique numeric token (e.g., `author_0`) is prepended to the input to specify the target author.	`<author_id>:` النص الأصلي باللغة العربية الفصحى
Descriptive Author Tokens with FT	Human-readable tokens (e.g., `<author:Yusuf_Idris>`) are used instead of numeric ones to improve interpretability.	`<author:name>:` النص الأصلي
Prompt Engineering with FT (Our Best System)	The task is framed as an explicit natural language instruction in Arabic, prepended to the input.	اكتب النص التالي بأسلوب `<author:name>: [source_text]`

Table 1: Overview of the four experimental approaches for authorship style transfer.

Approach	BLEU	chrF
Standard Fine-tuning	20.50	58.50
Numeric Tokens	24.04	59.15
Descriptive Tokens	24.00	59.00
Prompt Engineering	**24.58**	**59.01**

Table 2: Official results on the AraGenEval 2025 test set. Our prompt engineering system ranked first.

All code was executed on NVIDIA A100 GPUs (80GB VRAM) under CUDA 12.2. Models were fine-tuned with a batch size of 64 (across 4 GPUs), AdamW optimizer (weight decay 0.01), a 5×10^{-5} learning rate, and OneCycleLR scheduling with 1000 warmup steps. Training ran for up to 10 epochs with early stopping based on validation loss.

4.3 Generation Settings

Generation used beam search with sampling (`num_beams`=2, temperature=0.6, `top_k`=20, `top_p`=0.8, repetition penalty=1.05, length penalty=0.6) and a 512-token output cap to balance quality and diversity.

4.4 Evaluation Metrics

Evaluation was conducted using two metrics: **BLEU**, the primary measure of n-gram precision between generated and reference texts (Papineni et al., 2002), and **chrF**, a character n-gram F-score metric (Popović, 2015) often better suited for morphologically rich languages such as Arabic.

5 Results and Analysis

Our experimental results on the official test set clearly show the progression in performance across the four conditioning strategies. The prompt engineering approach achieved the highest scores, securing first place in the competition. The results are summarized in Table 2.

As shown in Table 2, explicit author conditioning was essential, with all conditioned methods outperforming the baseline. Human-readable tokens proved as effective as numeric ones, showing that interpretability does not reduce performance. Prompt engineering achieved the strongest results, enabling the model to leverage its pre-trained understanding of Arabic, while also reducing common errors such as semantic drift, incomplete style transfer, and repetition by better preserving entities and semantic fidelity.

5.1 Dataset Stylometric Analysis

We conducted a post-hoc stylometric analysis of the 21 authors using a custom `StylometricAnalyzer`, extracting lexical, syntactic, and statistically categorized features to create individual stylistic profiles. The resulting heatmap (Figure 2) revealed strong stylistic homogeneity, with minimal variation in core features like sentence length, vocabulary richness, complexity, and formality. Punctuation-based cues offered little discrimination, and the only notable outlier was ثروت أباظة, who showed lower emotional intensity—highlighting the challenge of style transfer in this dataset.

This observation provides a compelling explanation for the superior performance of our prompt engineering approach. Methods relying on implicit signals or simple author tokens must learn these subtle distinctions from the data alone. In contrast, the explicit instruction اكتب النص التالي بأسلوب leverages the vast, latent knowledge of the pre-trained AraT5 model. It effectively commands the

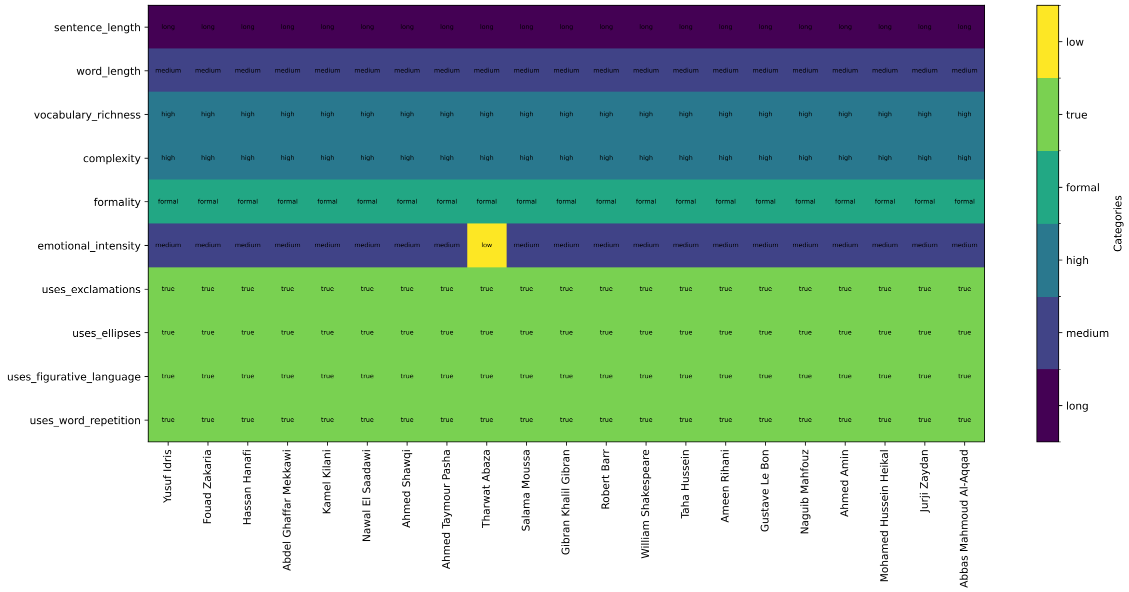


Figure 2: Stylometric characteristics heatmap.

model to access its deep understanding of authorial voice, which goes far beyond what our statistical metrics can measure. This allows it to capture the unique, nuanced characteristics of each author, leading to its first-place performance.

Task	Accuracy	F1
Authorship ID (Dev)	0.95	0.95
Authorship ID (Test)	0.87	0.83
ARATECT (Dev)	0.99	0.99
ARATECT (Test)	0.79	0.76

Table 3: Performance metrics for Subtasks 2 and 3.

6 Results in Additional Shared Tasks

6.1 Subtask 2: Authorship Identification

We addressed class imbalance through weighted loss during training. After preprocessing and tokenization, several Arabic-specific BERT-based models were fine-tuned. The best-performing configuration, `bert-base-arabic-camelbert-mix-sentiment` (Inoue et al.), trained for 10 epochs with early stopping, reached an accuracy of 95.3% and a macro F1-score of 95.1% on the development set. Our system ranked 6th, achieving an F1-score of 0.83138 and an accuracy of 87.52%, which is only 6.7 percentage points lower in F1-score compared to the top-ranked system (0.89886). The official leaderboard results for both subtasks are summarized in Table 3.

6.2 Subtask 3: ARATECT (Arabic AI-Generated Text Detection)

For AI-generated text detection, the dataset was already balanced. After minimal cleaning and tokenization, transformer-based models converged in just 3 epochs. Our top model, `XLM-RobertaForSequenceClassification` (Ruder et al., 2019), achieved an accuracy of 99.36% and a macro F1-score of 99.3% on the development set. Our system ranked 6th, achieving an F1-score of 0.76 and an accuracy of 79%, which is only 10 percentage points lower in F1-score compared to the top-ranked system (0.86).

7 Conclusion

In this paper, we presented our winning system for the AraGenEval 2025 Arabic Authorship Style Transfer task. Our systematic investigation demonstrates that explicit prompt engineering with natural Arabic instructions is a highly effective method for conditioning a T5 model. We found that simpler, interpretable conditioning methods are potent and that leveraging a model's linguistic capabilities through clear prompts yields superior results compared to merely adding special tokens. Future work could explore integrating stylometric features directly into the prompt, extending the framework to multi-author style transfer, and developing real-time applications. Our findings underscore the value of prompt engineering as a powerful and interpretable technique for controllable text generation in Arabic.

Acknowledgments

The authors gratefully acknowledge the support provided by Tuwaiq Academy and the computational resources by Prince Sultan University.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Association for Computational Linguistics.

Reem Alghamdi, Areej Al-Wabil, and Muna Al-Razgan. 2024. Distinguishing arabic genai-generated tweets and human tweets utilizing machine learning. *Engineering, Technology & Applied Science Research*, 14(5):16720–16726.

AbdelRahim Elmadany, Muhammad Abdul-Mageed, and 1 others. 2022. Arat5: Text-to-text transformers for arabic language generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers)*, pages 628–647.

Helena Gómez-Adorno, Juan-Pablo Posadas-Duran, Germán Ríos-Toledo, Grigori Sidorov, and Gerardo Sierra. 2018. Stylometry-based approach for detecting writing style changes in literary texts. *Computación y Sistemas*, 22(1):47–53.

Sol Horvitz, Luis Ortiz, Anjali Sharan, and Alexandra Getman. 2024. Tinystyler: Efficient few-shot text style transfer with authorship embeddings. *arXiv preprint arXiv:2406.15586*.

Zhumin Hu, Zhaofeng Tu, Zur G-BETH, Jdn Lrec, and Vıctor T-BI. 2022. Text style transfer: A review and experimental evaluation. *ACM SIGKDD Explorations Newsletter*, 24(1):1–21.

G Inoue, B Alhafni, N Baimukan, H Bouamor, and N Habash. The interplay of variant, size, and task type in arabic pre-trained language models. arxiv 2021. *arXiv preprint arXiv:2103.06678*.

AraGenEval Organizers. 2024. Overview of the arageneval 2024 shared task. Shared Task Website. (Placeholder citation).

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.

Krish Patel, Saizhu Zong, Vicky Shao, Ao Peng, and He He. 2022. Low-resource authorship style transfer: Can non-famous authors be imitated? *arXiv preprint arXiv:2212.08986*.

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.

Sebastian Ruder, Anders Søgaard, and Ivan Vulić. 2019. Unsupervised cross-lingual representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 31–38.

Vicky Shao, Xinyi Chen, Saizhu Zong, Yuerou Yang, Ao Peng, and He He. 2024. Authorship style transfer with inverse transfer data augmentation. *AI Open*, 5:14–22.

Athership at AraGenEval Shared Task: Identifying Arabic Authorship with a Dual-Model Logit Fusion

Eman Samir[*], **Mahmoud Rady**[*], **Maria Bassem**[*], **Mariam Hossam**[*],
Mohamed Amin[*], **Nisreen Hisham**[*], **Sara Gaballa**[*]
Applied Innovation Center (MCIT), Egypt
{e.samir, m.rady,m.bassem,mariam.hossam,
m.amin,n.hisham,s.gaballa}@aic.gov.eg
Ayman Khalafallah
ayman.khalafallah@alexu.edu.eg

Abstract

Authorship identification in Arabic is a challenging task due to the language's morphological richness, orthographic variation, and stylistic diversity across genres and authors. In this paper, we present our submission to Subtask 2: Authorship Identification of the AraGenEval 2025 Shared Task at ArabicNLP, which aims to identify the author of a given Arabic paragraph among a set of 21 authors. This task is important for applications such as digital forensics, plagiarism detection, literary analysis, and AI-generated content verification, where reliably linking text to its author can provide critical insights. We employ transformer-based encoders and address the dataset's class imbalance by leveraging an ensemble of two capable Arabic language understanding models: AraBERT and AraELECTRA. Our approach combines the pre-softmax logits of both models before the final softmax layer, effectively capturing complementary strengths in their predictions. Using our proposed method, we achieved third place on the Subtask 2 leaderboard of the AraGenEval Shared Task (Abudalfa et al., 2025), with a Macro-F1 score of 0.85968 and accuracy of 0.89516 on the test split.

1 Introduction

This paper details the system we developed for the AraGenEval 2025 Shared Task on Arabic Authorship and AI-Generated Text Detection, hosted at the Arabic Natural Language Processing Conference (ArabicNLP 2025) (Abudalfa et al., 2025). Our work is submitted under Subtask 2: Authorship Identification, a multi-class classification challenge designed to attribute a given Arabic text to its correct author from a closed set of 21 distinguished writers. The importance of this task has grown substantially with the proliferation of digital content. Robust authorship identification systems have critical real-world applications in digital forensics for identifying anonymous authors, in cybersecurity for detecting coordinated disinformation campaigns, in academic integrity for uncovering plagiarism, and in digital humanities for attributing disputed or anonymous literary works. The task is centered exclusively on the **Arabic language**, with a dataset curated to include diverse genres such as literary, philosophical, and journalistic prose, ensuring that solutions must focus on deep stylistic features rather than superficial topical cues.

The challenge of authorship attribution in Arabic is particularly acute due to the language's intrinsic complexities. Arabic is characterized by its **rich and complex morphology**, where a single root can spawn a vast array of words, making traditional bag-of-words models less effective. Furthermore, the phenomenon of **diglossia**—the coexistence of Modern Standard Arabic (MSA) with numerous regional dialects—means that authors often possess a unique stylistic blend, which may not be immediately apparent. Finally, **orthographic variability** in the Arabic script, such as the multiple forms of the hamza and the optionality of diacritics (*tashkeel*), introduces surface-level noise that can obscure an author's true stylistic signature. These linguistic hurdles are compounded by difficulties inherent in the dataset itself, including a notable **class imbalance** across the authors and significant stylistic diversity. Together, these complexities demand robust models capable of identifying an author's unique textual fingerprint amidst considerable noise.

To address these challenges, we fine-tuned two state-of-the-art Arabic Transformer encoders: **AraBERT** (Antoun et al.), trained with Masked Language Modeling (MLM), and **AraELECTRA** (Antoun et al., 2021), trained with Replaced Token Detection (RTD). Their complementary pretraining objectives were expected to cap-

[*] These authors contributed equally to this work.

ture different facets of authorial style. Our best system is a logit-level ensemble that averages the models' raw prediction scores before the softmax, leveraging their strengths and reducing individual weaknesses. We also tested a sliding-window strategy with AraBERTv02 for handling inputs longer than 512 tokens.

Our ensemble-based system, achieved **3rd place** in the final competition rankings, demonstrating its effectiveness on this challenging task. The key contributions and findings of our work can be summarized as follows:

- We demonstrate the successful application of fine-tuned AraBERT and AraELECTRA models for Arabic authorship attribution, using minimal preprocessing to ensure the preservation of subtle stylistic markers.
- We show that a logit-level ensemble of AraBERT and AraELECTRA significantly outperforms either model individually on both the development and final test sets, confirming the value of model fusion.
- We provide a valuable negative result from our sliding-window experiments with AraBERTv02, which indicates that simple chunking and aggregation for documents longer than 512 tokens degrades performance, highlighting the critical importance of contiguous context for stylistic analysis.
- We present a qualitative analysis, including correctly classified examples from stylistically complex passages, to illustrate the system's practical capabilities.

2 Related Work

Authorship attribution has evolved from early stylometric methods based on lexical and statistical features (Stamatatos, 2009) to modern deep learning approaches. For Arabic, traditional machine learning methods using character n-grams and morphological features (Shaker, 2017; Haddad et al., 2019) have shown promise but require extensive feature engineering. Neural models such as RNNs and CNNs (Alshahrani and Alsuhaymi, 2020) reduce this need, and transformer-based encoders like AraBERT (Abdul-Mageed et al., 2021) and AraELECTRA (Antoun et al., 2021) now achieve state-of-the-art results in Arabic NLP. Ensemble methods remain underexplored for Arabic authorship tasks, with only limited work in social media contexts (Alshehri and Al-Khazraji, 2022), despite evidence from other languages (Jafari Akinabad and Mohammadpour, 2021) that model combination can improve robustness. Our work fills this gap by applying a logit-level ensemble of AraBERT and AraELECTRA for literary and philosophical genres.

3 Dataset

The dataset was curated by the task organizers from 10 publicly available books for 21 authors. Books were segmented into semantically coherent paragraphs, yielding substantial variation in length and style. Table 1 summarizes the distribution of samples across train, validation, and test splits.

Author	**Train**	**Val**	**Test**
Ahmed Amin	2892	246	594
Ameen Rihani	1557	142	624
Hassan Hanafi	3735	548	1002
...	...	...	...
William Shakespeare	1236	238	358

Table 1: Example excerpt of dataset statistics; full table provided by organizers.

Paragraph lengths range from short excerpts of under 50 tokens to long passages exceeding the 512-token limit of standard Transformer models. The dataset is also **imbalanced**, with author sample counts ranging from a few hundred to several thousand, introducing a challenge for models to maintain performance on minority classes.

4 Methodology

4.1 Base Models: AraBERT and AraELECTRA

AraBERT is a 12-layer bidirectional Transformer encoder based on BERT (Devlin et al., 2019), pretrained on large-scale Arabic corpora (news, Wikipedia, social media) using the Masked Language Modeling (MLM) objective. This bidirectional training captures deep contextual relationships between words and morphemes, beneficial for Arabic's rich morphology. For our task, we add a linear classification layer on the final hidden state of the [CLS] token.

AraELECTRA follows the ELECTRA framework (Clark et al., 2020), replacing MLM with a Replaced Token Detection (RTD) objective, where the model discriminates between original and substituted tokens. This more sample-efficient training yields rich token-level representations. Architecturally, it is also a 12-layer Transformer encoder, with the same classification head as AraBERT.

We fine-tune both models for 4 epochs with a maximum sequence length of 512 tokens, truncating longer texts. This identical setup enables direct comparison and facilitates their combination in our logit-level ensemble.

4.2 Logit-Level Ensemble

Each model outputs logits $\ell^{(1)}, \ell^{(2)} \in R^{11}$. We combine them as:

$$\ell_{\text{ens}} = \ell^{(1)} + \ell^{(2)}, \quad p = \text{softmax}(\ell_{\text{ens}})$$

This preserves raw decision margins before applying the softmax.

4.3 Sliding-Window Experiment

We fine-tuned the BERT Large AraBERTv02 model (aubmindlab/bert-large-AraBERTv02) for authorship identification using a sliding-window approach to handle long paragraphs without losing context. Input texts were split into fixed-length sequences of 512 tokens (including special tokens) with a stride of 128 tokens, ensuring overlap between adjacent segments so that stylistic cues spanning boundaries were preserved.

The dataset was loaded from Excel files with author names label-encoded. To address class imbalance, balanced class weights were computed and passed to a custom Trainer subclass. We applied label smoothing with a factor of 0.1 to improve generalization.

At inference, document-level voting was implemented by aggregating chunk predictions to produce the final author label.

4.4 Baselines

- TF-IDF + FCN: Character and word n-gram features via TF-IDF, fed into a 2-layer fully connected network.

- Contrastive (Qarib) + k-NN: Contrastive learning on Qarib (Abdelali et al., 2021) encoder embeddings to bring same-author texts closer in vector space, followed by k-nearest neighbors classification.

4.5 Negative Experiment: Simple Chunking

We attempted to split long texts (>512 tokens) into smaller chunks (512 and remainder), assigning the same label to all chunks. This degraded accuracy, likely because shorter fragments sometimes lack sufficient stylistic cues.

5 Results

Table 2 summarizes the performance of our models on the development set. Among the individual models, **AraBERT** achieved the highest development accuracy (0.90) and a Macro-F1 score of 0.84, slightly outperforming AraELECTRA (0.88 accuracy, 0.83 Macro-F1). Our logit-level **ensemble** of AraBERT and AraELECTRA produced the best overall results on the development set, with an accuracy of 0.92 and a Macro-F1 score of 0.86, confirming the benefit of combining the two architectures.

We also evaluated several alternative approaches. A **sliding-window** inference strategy applied to AraBERTv02 slightly improved the Macro-F1 score over the single-model baselines (0.85) but did not surpass the ensemble. Traditional **TF-IDF** features followed by a fully connected network (FCN) performed considerably worse (approximately 0.75 accuracy, 0.70 Macro-F1), highlighting the limitations of shallow lexical representations for this task. A **contrastive learning** approach achieved moderate performance (0.84 accuracy, 0.79 Macro-F1), suggesting that more specialized contrastive objectives might be needed for stylistic analysis.

Our final submission to the AraGenEval 2025 Subtask 2 leaderboard achieved a Macro-F1 score of 0.85968 and an accuracy of 0.89516 on the held-out test set, placing third overall in the competition. These results demonstrate the effectiveness of our ensemble strategy in capturing complementary stylistic cues from the two pretrained models.

To illustrate the system's ability to capture nuanced stylistic patterns, we present two correctly classified examples from the test set:

Example 1 — Philosophical Prose:
Input excerpt: وهنا تصبح الطبيعة في حاجة إلى مبرر وهكذاً يقدم كانت فرضًا تفسيريًا محضًا لا يغير من محتوى المعرفة.

Model	Dev Acc.	Dev Macro F1
AraBERT	0.90	0.84
AraELECTRA	0.88	0.83
Ensemble	0.92	0.86
Sliding Window	0.90	0.85
TF-IDF + FCN	∼0.75	0.70
Contrastive	0.84	0.79

Table 2: Model performance on the development set.

Predicted author: فؤاد زكريا *(correct)*

Example 2 — Literary Prose:
Input excerpt: ثمن الكتابة… لا أجيد كتابة المقدمات، يمكن أن أكتب قصةً من ألف صفحة… يدق بالمطرقة على جواز سفرها فتدخل.

Predicted author: نوال السعداوي *(correct)*

6 Discussion

The experimental results indicate that the ensemble of AraBERT and AraELECTRA consistently outperformed either model individually on both the development and test sets. We attribute this improvement to the complementary nature of the models' pretraining objectives: AraBERT's masked language modeling encourages deeper bidirectional context modeling, while AraELECTRA's replaced token detection promotes fine-grained token-level discrimination. By combining their pre-softmax logits, the ensemble is able to integrate these distinct strengths, leading to more robust stylistic representation and classification.

The limited gains observed from the sliding-window approach suggest that splitting long texts into chunks may disrupt important discourse-level cues, which are often essential for capturing an author's style. Similarly, the relatively low performance of the TF-IDF + FCN baseline confirms that surface lexical features alone are insufficient for distinguishing between highly skilled Arabic authors with overlapping vocabularies. The moderate results of the contrastive learning approach point to the need for more task-specific contrastive objectives that explicitly model stylistic similarity and difference.

Overall, the findings highlight the value of leveraging multiple pretrained encoders with different inductive biases, while also underscoring the importance of preserving global context in Arabic authorship attribution tasks.

7 Conclusion and Future Work

In this paper, we presented a logit-level ensemble of AraBERT and AraELECTRA for Arabic authorship attribution, developed for the AraGenEval 2025 Shared Task. Our approach leveraged the complementary strengths of two transformer-based encoders with different pretraining objectives, resulting in robust performance across literary, philosophical, and journalistic genres. The system achieved third place on the competition leaderboard, with a Macro-F1 score of 0.85968 and an accuracy of 0.89516 on the held-out test set. The results demonstrate that combining pretrained models is an effective strategy for addressing the linguistic and stylistic challenges of Arabic authorship identification.

For future work, we plan to extend our ensemble in two directions. First, we will explore *weighted* logit-level fusion, where the contribution of each model is learned or tuned based on validation performance rather than averaged equally. Second, we aim to increase the number of diverse models in the ensemble, incorporating additional pretrained Arabic encoders and possibly multilingual transformers. We expect that both strategies will further enhance performance by capturing a wider range of stylistic and contextual features, thereby improving the system's robustness and generalization.

Acknowledgments

We would like to thank the organizers of the AraGenEval 2025 Shared Task for providing the dataset and evaluation platform, and for their efforts in fostering research on Arabic NLP. Finally, we are grateful to our colleagues and peers for their valuable feedback during the development of this work.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

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.

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*. Association for Computational Linguistics.

Nourah Alshahrani and Hadeel Alsuhaymi. 2020. Deep learning for arabic authorship attribution. In *2020 6th International Conference on Information Management (ICIM)*, pages 8–14. IEEE.

Nourah Alshehri and Mohammed Al-Khazraji. 2022. Ensemble methods for arabic author profiling on social media data. *Future Internet*, 14(2):51.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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.

Ramy Haddad, Wael Obeid, and Hani Jundi. 2019. Morphological features for arabic authorship attribution. In *International Conference on Information and Communication Technologies for Development*, pages 615–624. Springer.

Zahra Jafari Akinabad and Masoud Mohammadpour. 2021. Ensemble learning methods in authorship attribution. In *2021 7th International Conference on Web Research (ICWR)*, pages 1–6. IEEE.

Zaid Shaker. 2017. Arabic authorship identification using n-gram and support vector machine. In *International Conference on Advanced Machine Learning Technologies and Applications*, pages 507–516. Springer.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

A Appendix

A.1 Hyperparameter Settings

Table 3 lists the main hyperparameters used for fine-tuning AraBERT and AraELECTRA in our experiments.

Hyperparameter	**Value**
Optimizer	AdamW
Learning rate	2×10^{-5}
Batch size	16
Weight decay	0.01
Epochs	4
Max sequence length	512

Table 3: Hyperparameters for fine-tuning.

A.2 Hardware and Runtime

All experiments were run on a single NVIDIA P100 GPU with 16GB of memory. Fine-tuning each model for 4 epochs required approximately 3 hours.

Sebaweh at AraGenEval Shared Task: BERENSE - BERt based ENSEmbler for Arabic Authorship Identification

Muhammad Helmy*
mu.helmy@nu.edu.eg

Batool Balah*
batoolnajeh@gmail.com

Ahmed Mohamed Sallam
ahmedm.sallamibrahim@gmail.com

Ammar Sherif
ammarsherif90@gmail.com

Abstract

Authorship Identification for Arabic texts is challenging due to the language's dialectal diversity and the wide stylistic variation across genres, cultures, and historical periods. It has critical applications in copyright enforcement, forensic linguistics, and literary analysis. Recognizing its importance, we addressed this challenge using the *AraGenEval 2025* shared task dataset, which contains works by writers from diverse backgrounds and time periods. We conducted extensive experiments with multiple architectures and proposed an ensemble model that combines the strengths of four fine-tuned transformer-based models. We applied data augmentation to enrich the dataset and class weighting to handle class imbalance during training. Our system achieved a **Macro-F1 score of 90%**, representing a **15% improvement** over our baseline, and ranked **1st** in the competition.

1 Introduction

Transformer architectures have revolutionized the way we analyze and understand textual data, demonstrating a remarkable ability to capture deep contextual and stylistic patterns highly effective for tasks such as Authorship Identification. This task involves determining the author of a given text based on its stylistic and linguistic characteristics and has critical applications in plagiarism detection, forensic linguistics, and historical literature analysis. However, Arabic remains underrepresented in this line of research, despite its rich literary tradition (Alqurashi, 2024).

The task presents four core challenges: language-related complexities, feature selection, data availability, and preprocessing decisions. The structural challenges of Arabic, such as morphological richness, inflection, diglossia, and diacritics, complicate preprocessing and obscure stylistic cues. Additionally, the scarcity of large, balanced corpora and suitable modeling tools further hinders progress (Alqahtani and Dohler, 2023).

Our main contributions to the Arabic Authorship Identification task:

- Ranked 1st in AraGenEval's Subtask 2 on Arabic Authorship Identification (Abudalfa et al., 2025), a multiclass classification task predicting the author of an Arabic paragraph.
- Performed data augmentation to enrich the samples of underrepresented authors and applied class weighting during training.
- Extensively experimented with multiple Arabic transformer models (Alqurashi, 2024; Alqahtani and Dohler, 2023) and combined them into an ensemble, which reduced variance and improved robustness.
- Achieved a +15% improvement in macro-averaged F1 over the baseline, reaching 90%.

2 Background

The dataset for AraGenEval's Subtask 2 includes 21 Arabic authors spanning novelists, philosophers, historians, social activists, and politicians, and covers diverse time periods. Each author is represented by one to ten books, segmented into semantically coherent paragraphs. The texts are exclusively in Arabic, encompassing Classical Arabic, Modern Standard Arabic (MSA), and Egyptian dialect. Class distributions vary widely, from fewer than 100 to over 3000 samples per author, reflecting real-world authorship identification challenges such as long-form input, class imbalance, genre variability, and subtle stylistic overlap.

Authorship identification in English has evolved from classical machine learning with handcrafted features to deep learning and transformer-based approaches. Huertas-Tato et al. (Huertas-Tato et al.,

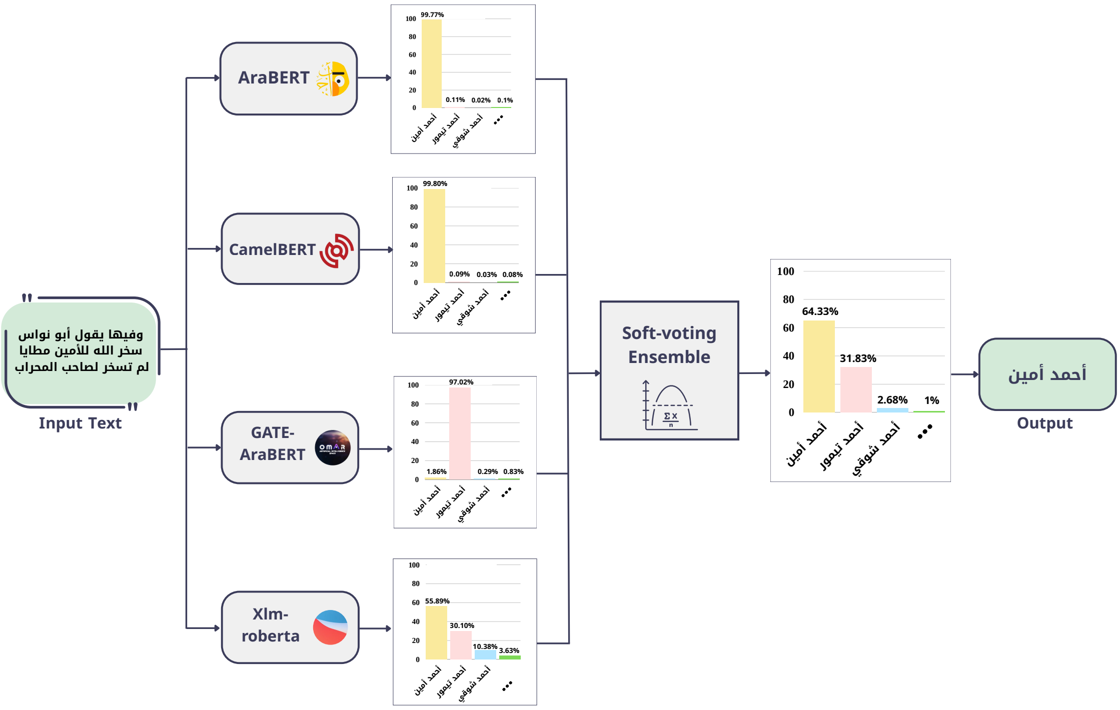


Figure 1: **System overview**. Our system ensemble is composed of 4 models: AraBERT, CAMeLBERT, XLM-RoBERTa-Arabic, and GATE-AraBERT-v1. The final output is then computed via soft-voting of all the outputs.

2022) introduced PART, a pre-trained transformer using contrastive learning to capture author-specific styles. Silva et al. (Silva et al., 2023) applied GAN-BERT to attribute late 19th-century novels and later extended it to detect AI-generated forgeries (Silva et al., 2024). While highly effective across genres and large author sets, comparable work in Arabic remains scarce due to its morphological richness and dialectal variation, which both complicate modeling and offer unique stylistic cues.

A related task, Author Profiling, predicts attributes such as gender, dialect, or age. Zhang and Abdul-Mageed (Zhang and Abdul-Mageed, 2022) developed a transformer-based system for profiling Arabic social media users. However, such work focuses on trait prediction for short, informal texts, not full-text identity attribution, highlighting the need for dedicated Arabic authorship identification methods across domains.

Arabic authorship studies have often been small-scale (fewer than 15 authors) and domain-specific, such as classical literature, Islamic legal texts, or poetry. These works aimed to identify authors using statistical and machine learning methods adapted to the domain. Al-Sarem et al. (Al-Sarem et al., 2020) used an artificial neural network for fatwa texts, while Sayoud (Hadjadj and Sayoud, 2021) applied PCA and SMOTE to address feature dimensionality and class imbalance. Earlier works (Altheneyan and Menai, 2014; Ahmed et al., 2019) employed Naïve Bayes, SVM, or LDA with lexical, syntactic, and structural features. While effective in restricted settings, these approaches relied heavily on manual feature engineering and often failed to capture semantic or stylistic depth across genres.

More recent Arabic work with transformers remains narrow in scope. AlZahrani and Al-Yahya (AlZahrani and Al-Yahya, 2023) focused on Islamic legal texts with small author sets, while Alqurashi et al. (Alqurashi et al., 2025) used a CAMeLBERT-based ensemble for classical poetry, achieving F1 scores from 0.97 to 1.0. Despite strong results, their focus was limited to a single genre.

To address these gaps, our work presents a transformer-based model trained on Arabic texts spanning diverse dialects and genres, capable of learning stylistic patterns directly from raw text without manual feature engineering.

3 System Overview

We reached this system design after experimenting with several alternative architectures, including BERT embeddings with RNN/LSTM heads, frozen BERT embeddings with SVM/RF classifiers, and BERT embeddings concatenated with extracted topic distributions followed by a fully connected softmax layer. However, the pure BERT embeddings followed by a fully connected softmax layer outperformed the other approaches (see Figure 1).

3.1 Model Architecture

Following the best-performing architecture, we fine-tuned four transformer-based models from Hugging Face: AraBERT v0.2 (136M), CAMeLBERT-Mix (110M), Arabic XLM-RoBERTa (270M), and GATE-AraBERT (135M), each leveraging the same fully connected softmax classification head. To ensure robust inference, we employed a soft-voting ensemble that averaged the predicted probability distributions of all four models, thus reducing variance and exploiting complementary stylistic features captured by each transformer (see Appendix B).

3.2 Handling Class Imbalance

The dataset exhibited a significant imbalance in the number of samples per author, which could bias the model toward overrepresented classes. To address this, we modified the standard cross-entropy loss to include class weights inversely proportional to class frequencies, thereby penalizing errors on underrepresented authors more heavily (see Appendix C for the formal definition).

3.3 Data Augmentation

To increase stylistic variation and expand data diversity, we collected additional works from the Hindawi Books dataset (Filali, 2022), targeting underrepresented authors: Tharwat Abaza, Kamel Kilani, Gobran Khalil Gobran, Ahmad Taymour Basha, Ahmad Shawqy. After using the validation set to select the hyper-parameters and do initial experiments, we appended it with the training set at the end to increase the training data before the final evaluation on the test set.

4 Experimental Setup

4.1 Data Splits

We followed the official Shared Task 2 data split provided by the organizers. The dataset was divided into *training*, *validation*, and *test* sets. The validation set was used for model selection and hyperparameter tuning, while the test set was reserved for final evaluation.

4.2 Preprocessing

To address statistical imbalances and reduce noise that could obscure stylistic cues, we applied three preprocessing steps to the dataset. First, we removed a total of 2,740 duplicates to avoid overrepresentation of specific expressions. Second, we performed length capping by splitting 1,381 texts exceeding 3,000 characters into chunks of approximately 2,000 characters, corresponding to the mean text length across authors and remaining within the tokenizer's maximum sequence length. (see Appendix D for illustrative examples).

This step was intended to reduce overfitting risks, improve gradient updates for underrepresented authors, and encourage reliance on stylistic rather than length cues. Finally, we removed diacritics, as they are often inconsistently applied or auto-inserted in digital-born text, which can introduce noise into the stylistic signal.

4.3 Parameter Settings

We fine-tuned four transformer-based models with carefully selected hyperparameters, including learning rate, optimizer, training epochs, warmup ratio, and weight decay. The best configurations for AraBERT, CAMeLBERT, and XLM-RoBERTa-Arabic are the same: learning rate of $8e10^{-5}$, Adam as optimizer, cosine scheduler, 10% warmup ratio, 4 epochs, and 0.1 of weight decay. GATE-AraBERT-v1 is the same with the only difference in learning rate: $2e10^{-5}$

4.4 External Tools and Libraries

The implementation was carried out in Python 3.10 using Google Colab and Kaggle environments. We used **pandas** and **numpy** for data handling, **matplotlib** and **seaborn** for visualization (e.g., histograms and bar charts), **langdetect** for language identification, and **langchain** for text splitting.

4.5 Evaluation Metrics

Following the AraGenEval guidelines, we evaluated our models using four primary metrics: Macro F1-score, Accuracy, Precision, and Recall on the test set. Macro F1-score was the main ranking criterion in the shared task, defined as:

$$\text{Macro F1} = \frac{1}{N} \sum_{i=1}^{N} \text{F1}_i \tag{1}$$

where N is the number of classes, and F1_i is the F1-score computed for class i:

$$\text{F1}_i = 2 \frac{\text{Precision}_i \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{2}$$

$$\text{where Precision}_i = \frac{TP_i}{TP_i + FP_i}, \tag{3}$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \tag{4}$$

Here, TP_i, FP_i, and FN_i denote the number of true positives, false positives, and false negatives for class i. Accuracy is computed as the proportion of correctly predicted instances over the total number of instances.

5 Results

We gradually enhanced performance over our initial BERT + RNN baseline. Table 1 compares alternative architectures we tested. The best single-model result came from BERT embeddings with a softmax layer, reaching **0.85**. This suggests that while BERT embeddings capture valuable stylistic information, and their effectiveness depends heavily on the classifier's capacity to exploit high-dimensional contextual features.

Table 1: Comparison of alternative architectures on the validation set.

Architecture	F1 Score
BERT + RNN (baseline)	0.75
Frozen BERT + SVM (bagging)	0.66
Frozen BERT + Random Forest	0.35
BERT + Fully Connected Layer	**0.85**
Our Ensemble[1]	**0.90**

Building on these findings, we adopted the BERT embeddings + fully connected softmax layer architecture as our main design and explored further enhancements. We evaluated various embedding models, including AraBERT v0.2, CAMeLBERT-Mix, Arabic XLM-RoBERTa, GATE-AraBERT, Arabic-labse-Matryoshka, and Arabic distilbert-base. We excluded the last two from the final ensemble as their validation F1 scores fell below **0.80**.

We incorporated external stylistic cues by performing topic modeling and concatenated the top topic keywords with the embedding representation, following the approach of Alqurashi et al. (Alqurashi et al., 2025). However, experiments with CAMeLBERT-Mix showed no measurable performance gain (F1 = **0.85** both with and without topic features), suggesting that topic distributions did not contribute additional discriminative power beyond the contextual embeddings.

Subsequently, augmenting training data with the Hindawy dataset yielded consistent validation improvements across most models. Table 2 reports macro-F1 scores with and without augmentation on the validation set.

Table 2: Macro-F1 with and without augmentation (validation set).

Model	Aug	No Aug
AraBERT v0.2	0.90 (↑ **2**%)	0.88
CAMeLBERT-Mix	0.90 (↑ **6**%)	0.84
Arabic XLM-RoBERTa	0.83 (**0**)	0.83
GATE-AraBERT	0.89 (↑ **5**%)	0.84

Although applying class-weighted loss improved performance in the frozen GATE-AraBERT + bagging SVM setup, increasing validation F1 from **0.56** to **0.66**, it did not show such an enhancement for the fully connected architecture. The effect was minimal overall, though we observed a slight gain from **0.82** to **0.83** validation F1 for XLM-RoBERTa. We retained this procedure as it did not degrade performance for other models and XLM-RoBERTa had not shown improvements from data augmentation.

To better understand model errors, we inspected the confusion matrix of the predicted authors. Misclassifications were often concentrated among authors with overlapping genres or historical contexts, reflecting the stylistic and thematic proximity between them. A detailed analysis of the most frequent confusions is provided in Appendix A.

Finally, our ensemble system achieved a macro-averaged F1 of **0.9046**, accuracy of **0.9327**, precision of **0.9012**, and recall of **0.9143**, ranking **1st** on the official test set of the AraGenEval 2025 Subtask 2, outperforming each single model.

6 Conclusion

We developed an ensemble-based system for Arabic Authorship Identification, achieving a macro-F1 of **0.9046** on the AraGenEval 2025 test set and

[1] Result on test set.

ranking **1st** in Subtask 2. Our analysis showed that while frozen embeddings with classical classifiers underperformed, a BERT + fully connected design, combined with data augmentation and ensembling, delivered strong gains. Class-weighted loss had mixed effects, benefiting some models but not others.

Limitations include the restriction to only 21 authors and the features are not guaranteed to be style-based rather than content-based, which might present a form of overfitting. Future work will investigate open-set authorship, experiment more with contrastive learning to enhance the features, assess potential data leakage, and apply interpretability techniques to better understand the model's decision-making process.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The arageneval shared task on arabic authorship style transfer and ai-generated text detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

A. Ahmed, R. Mohamed, and B. Mostafa. 2019. Arabic poetry authorship attribution using machine learning techniques. *Journal of Computer Science*, 15(7):1012–1021.

Mohammed Al-Sarem, Abdullah Alsaeedi, and Faisal Saeed. 2020. A deep learning-based artificial neural network method for instance-based arabic language authorship attribution. *International Journal of Advances in Soft Computing and its Applications*, 12:1.

Fatimah Alqahtani and Mischa Dohler. 2023. Survey of authorship identification tasks on arabic texts. *ACM Computing Surveys*.

Lama Alqurashi. 2024. *Investigating Authorship in Classical Arabic Poetry Using Large Language Models*. Ph.D. thesis, University of Leeds.

Lama Alqurashi, Serge Sharoff, Janet Watson, and Jacob Blakesley. 2025. Bert-based classical arabic poetry authorship attribution. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6105–6119, Abu Dhabi, UAE. Association for Computational Linguistics.

Alaa Saleh Altheneyan and Mohamed El Bachir Menai. 2014. Naïve bayes classifiers for authorship attribution of arabic texts. *Journal of King Saud University - Computer and Information Sciences*, 26(4):473–484. Special Issue on Arabic NLP.

Fetoun Mansour AlZahrani and Maha Al-Yahya. 2023. A transformer-based approach to authorship attribution in classical arabic texts. *Applied Sciences*, 13(12).

Ali El Filali. 2022. Hindawi books dataset. `https://huggingface.co/datasets/alielfilali01/Hindawi-Books-dataset`. Accessed: 2025-08-10.

Hassina Hadjadj and H. Sayoud. 2021. Arabic authorship attribution using synthetic minority oversampling technique and principal components analysis for imbalanced documents. *International Journal of Cognitive Informatics and Natural Intelligence*, 15(1):1–17.

Javier Huertas-Tato, Álvaro Huertas-Garcia, Alejandro Martin, and David Camacho. 2022. Part: Pretrained authorship representation transformer. *arXiv preprint arXiv:2209.15373*. Preprint.

Kanishka Silva, Burcu Can, Frédéric Blain, Raheem Sarwar, Laura Ugolini, and Ruslan Mitkov. 2023. Authorship attribution of late 19th century novels using gan-bert. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Student Research Workshop), Volume 4*, pages 310–320. Association for Computational Linguistics.

Kanishka Silva, Ingo Frommholz, Burcu Can, Frédéric Blain, Raheem Sarwar, and Laura Ugolini. 2024. Forged-gan-bert: Authorship attribution for llm-generated forged novels. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 325–337. Association for Computational Linguistics.

Chiyu Zhang and Muhammad Abdul-Mageed. 2022. Bert-based arabic social media author profiling. *arXiv preprint arXiv:1909.04181*.

A Detailed Error Analysis

Inspection of the confusion matrix of the predicted authors revealed that **Tharwat Abaza** was often misclassified as **Ahmad Shawqi** and **Mohamed Hussein Heikal** due to narrative similarities. **Fouad Zakaria** and **Abd al-Ghaffar Mikkawi** occasionally confused, likely due to shared philosophical themes.

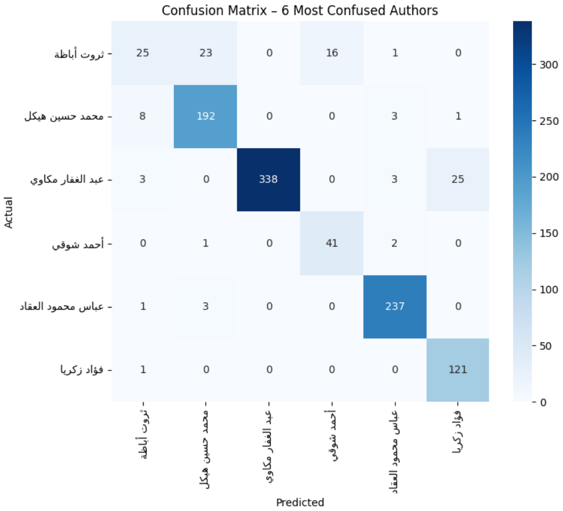


Figure 2: Confusion matrix showing frequent misclassifications between authors with overlapping styles.

B Soft-Voting Ensemble

In the soft-voting ensemble, the class probability distributions predicted by each model are averaged before selecting the final class label. Formally, let $p^{(m)} \in R^K$ denote the probability vector predicted by model m over K classes, and let M be the total number of models. The ensemble probability distribution $\hat{p}$ and the final predicted label $\hat{y}$ are defined as:

$$\hat{p} = \frac{1}{M} \sum_{m=1}^{M} p^{(m)}, \quad \hat{y} = \arg\max_k \hat{p}_k$$

where $\hat{p}$ represents the averaged probability distribution and $\hat{y}$ is the predicted class corresponding to the maximum probability.

C Weighted Loss Function

Formally, let $y_i \in \{1, \ldots, K\}$ denote the true class label of the i-th sample, $p_{i,c}$ the predicted probability for class c, and w_c the weight assigned to class c. The weighted cross-entropy loss is given by:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} w_{y_i} \log p_{i,y_i}$$

where N is the number of training samples and K the number of classes.

The weights w_c are set inversely proportional to the class frequencies, following the "balanced" option in `sklearn.compute_class_weight`:

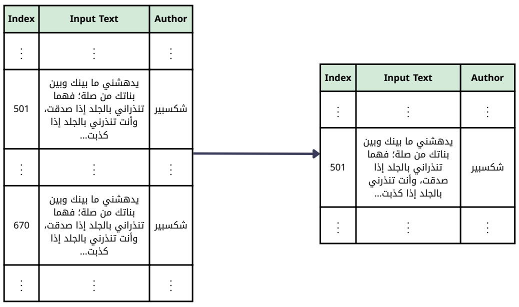
$$w_c = \frac{N}{K \cdot n_c},$$

where n_c is the number of samples belonging to class c. This ensures that underrepresented classes receive higher weights during training.

D Preprocessing Examples

Duplicate Removal

The following excerpt, shown in Figure 3, appeared multiple times in the dataset and was reduced to a single occurrence during preprocessing:



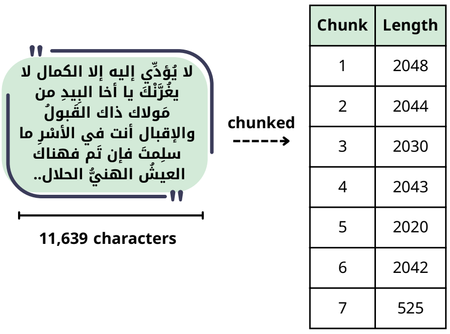
Index	Input Text	Author
⋮	⋮	⋮
501	يدهشني ما بينك وبين بناتك من صلة؛ فهما تنذراني بالجلد إذا صدقت، وأنت تنذرني بالجلد إذا كذبت...	شكسبير
⋮	⋮	⋮
670	يدهشني ما بينك وبين بناتك من صلة؛ فهما تنذراني بالجلد إذا صدقت، وأنت تنذرني بالجلد إذا كذبت...	شكسبير
⋮	⋮	⋮

Index	Input Text	Author
⋮	⋮	⋮
501	يدهشني ما بينك وبين بناتك من صلة؛ فهما تنذراني بالجلد إذا صدقت، وأنت تنذرني بالجلد إذا كذبت...	شكسبير
⋮	⋮	⋮

Figure 3: Example of a duplicate sample being reduced to one unique sample.

Splitting Large Texts

Figure 4 illustrates how a long text of 11,639 characters was split into seven smaller chunks of approximately 2,000 characters each, respecting the tokenizer's maximum input length.



لا يُؤدِّي إليه إلا الكمال لا
يغُرَّنْكَ يا أخا البِيدِ من
مَولاك ذاك القَبولُ
والإقبال أنت في الأسْرِ ما
سلِمتَ فإن تَم فهناك
العيشُ الهنيُّ الحلال..

Chunk	Length
1	2048
2	2044
3	2030
4	2043
5	2020
6	2042
7	525

Figure 4: Example of length splitting: a long text was divided into seven chunks with sizes [2048, 2044, 2030, 2043, 2020, 2042, 525].

CUET-NLP_Team_SS306 at AraGenEval Shared Task: A Transformer-based Framework for Detecting AI-Generated Arabic Text

Sowrav Nath[*], **Shadman Saleh**[*], **Kawsar Ahmed**
and Mohammed Moshiul Hoque
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
{u2004006, u2004030, u1804017}@student.cuet.ac.bd
moshiul_240@cuet.ac.bd

Abstract

With the rapid emergence of large language models (LLMs), AI-generated content has increased, presenting new opportunities and significant risks. Detecting such content is crucial, yet while research in high-resource languages like English has advanced, work in low-resource languages, such as Arabic, remains limited. To help fill this gap, the AraGenEval 2025 workshop organized a shared task on AI-generated Text Detection in Arabic. We participated in Task 3, where we evaluated several transformer-based models, including AraBERT, RoBERTa, AraRoBERTa, mBERT, and marBERT, both with and without chunking of input sequences during training. The experimental results show that applying chunking prior to training improves the performance of transformers. Among the evaluated models by the system testset, AraBERT with chunking achieved the highest F1 score (0.67), outperforming the others. Based on these results, our team ranked 12th in Shared Task 3.

1 Introduction

The rise of large language models (LLMs) has transformed text production, enabling rapid generation of coherent, human-like content. This evolution presents opportunities in creative writing, software engineering, and customer support, but also introduces risks to the integrity of educational assessment. Additionally, LLMs can enhance the sophistication and accessibility of social engineering attacks in online communication, leading to more convincing scams and the dissemination of misinformation. Reliable detection of AI-generated text is essential for maintaining trust and authenticity. While advances have occurred for languages such as English, Arabic remains challenging due to its characteristics, including root-and-pattern word construction, inflectional complexity, diverse dialects, and diacritics. Consequently, systems developed for high-resource languages frequently underperform when handling Arabic.

This work addresses these critical gaps, motivated by the need for reliable AI-generated text detection tools tailored to Arabic. We evaluate various transformer-based models for this purpose as part of the **AraGenEval 2025** shared task (Abudalfa et al., 2025). We investigate transformer-based models, including AraBERT(Antoun et al., 2020), RoBERTa (Liu et al., 2019), mBERT (Devlin et al., 2019) etc, both with and without chunking of input sequences. This work aims to provide insights into the strengths of current techniques and highlight the specific challenges of detecting AI-generated text in Arabic. The key contributions in this work are as follows:

- Evaluated multiple transformer-based models for detecting AI-generated text in Arabic.
- Introduced a chunking and confidence base aggregation approach with transformers to enhance detection performance.

2 Background

While most work in detecting machine-generated text has been conducted in high-resource languages (HRLs), such as English, some efforts have begun in low-resource languages (LRLs), including Arabic. Prova, 2024 made significant efforts to detect AI-generated text using BERT (Devlin et al., 2019), XGB (Chen and Guestrin, 2016), and SVM techniques. BERT models performed the best in the task, achieving an F1 score of 0.93. However, the research focused on English. Recent work by (Zhang et al., 2024) proposes a novel approach to distinguish between human and AI text. They integrated traditional TF-IDF (Takenobu, 1994) strategies with machine learning algorithms like Bayesian classifiers, Stochastic Gradient Descent

[*] Authors contributed equally to this work.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 65–71
November 8-9, 2025 ©2025 Association for Computational Linguistics

(SGD), and Categorical Gradient Boosting (CatBoost) (Prokhorenkova et al., 2019). Their methods reached an impressive ROC-AUC score of 0.975 on English text. In another study (Sadasivan et al., 2025), several types of detectors were assessed, including watermarking, neural network-based detectors, zero-shot detectors, and retrieval-based detectors. They found that AI detectors can be fooled by recursive paraphrasing, meaning the text is repeatedly reworded to evade detection. One major issue with Arabic language detection is handling diacritics, which are marks used in Arabic script to indicate pronunciation. Recent work by (Alshammari and Elleithy, 2024) focused on this challenge, comparing transformer-based models such as AraELECTRA (Antoun et al., 2021), AraBERT (Antoun et al., 2020), XLM-R (Conneau et al., 2020), and mBERT (Devlin et al., 2019). They showed that AI-detection systems struggle with Arabic text that includes diacritics and often misclassify human-written text as AI-generated.

Similar challenges exist for other LRLs. For example, a study on AI-generated review classification in Malayalam (Hasan et al., 2025) used LLMs to identify AI-generated reviews. The Gemma-2B model achieved an F1-score of 0.89. This demonstrates the potential of LLMs in detecting AI-generated content in underrepresented languages. With these findings in mind, this work employed preprocessing steps in which diacritics were removed and variants of Arabic letters were normalized. Subsequently, transformer-based techniques were applied to detect AI-generated text. In contrast to previous studies that primarily focused on HRLs or the role of diacritics in Arabic, this work utilizes chunking of input sequence before training and confidence based aggregation in output with transformer-based models to enhance long-context representation in Arabic AI-text detection.

3 Dataset and Task Description

The shared task[1], ARATECT: Arabic AI-Generated Text Detection, was part of the AraGenEval (Abudalfa et al., 2025) challenge. It focuses on distinguishing between human-written and AI-generated Arabic text. The ARATECT dataset comprises two primary sources. First, human-written texts were collected from reputable Arabic news sites and verified literary sources. Second, AI-generated texts were produced using Arabic-compatible large language models (e.g., GPT-4, Mistral, LLaMA) through diverse prompting strategies. Participants received a labeled training set of Arabic text samples with binary labels (human or machine). They also received an unlabeled test set for evaluation. The training set contains 4,798 samples (2,399 per class), and the test set includes 500 unlabeled samples, as shown in Table 1. The task was hosted on Codabench [2]. It aimed to advance Arabic AI-generated content detection.

Set	**Class**	S_C	A_W	**Min**	**Max**	T_S
Train	Human	2399	657	1	3068	54839
	Machine	2399	314	9	1969	37768
Test	All	500	230	12	1589	7772

Table 1: ARATECT dataset statistics. S_C: sample count, A_W: average words per sample, Min/Max: minimum and maximum words per sample, and T_S: total sentences.

4 System Overview

Several transformer models are implemented with and without the chunking of input sequence before training and investigated to address the tasks. Figure 1 outlines the methodology.

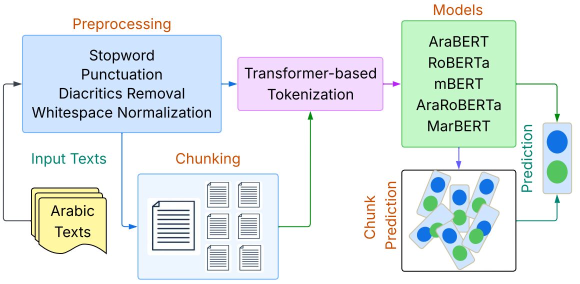


Figure 1: Schematic process of Arabic AI-generated content detection.

4.1 Data Preprocessing

Several preprocessing steps were applied to prepare the dataset for model training. For the training data, each sample was made using only the content field. For the test data, the title and content fields were concatenated. Subsequent preprocessing involved removing diacritics and normalizing variant Arabic letters. Repeated characters were eliminated using regular expressions. In addition, non-essential punctuation and special characters were removed. Excessive whitespace was normalized. Finally, labels were mapped to binary values in the training dataset.

[1] https://ezzini.github.io/AraGenEval/

[2] https://www.codabench.org/competitions/9120/

4.2 Transformer-based Models

Transformer-based models were used for this task because they efficiently process large-scale contextual information, making them well-suited for multilingual text classification. Several pre-trained transformer models from Hugging Face, including RoBERTa (Liu et al., 2019) and AraBERT (Antoun et al., 2020), mBERT (Devlin et al., 2019), ara-RoBERTa (Liu et al., 2019), MarBERT (Abdul-Mageed et al., 2020) were evaluated. Before passing data through the transformers, preprocessing and tokenization were done using each model's respective tokenizer. Inputs were truncated or padded to a maximum sequence length of 512. Since many input texts exceeded this maximum length, we applied a chunking strategy. Specifically, long texts were split into overlapping chunks of 400 with an overlap of 50 to preserve contextual continuity across chunks. Each chunk was independently processed through the model to obtain a confidence score. To aggregate predictions, we grouped chunks based on their original document IDs and computed the mean confidence score across all chunks. The final classification label was then derived from this aggregated score. This averaging approach ensures that information from all parts of the input sequence is considered, rather than being biased toward the first 512 tokens, thereby making the model more robust to long and information-dense texts. A formal description of the chunking and aggregation method is provided in Appendix A while Appendix A.5 reports rationale behind the choice of chunk size of 400 with overlap of 50.

Parameter	Value
Batch Size	16
Epochs	5
Weight Decay	0.001
Learning Rate	2e-5

Table 2: Hyperparameter configuration for the transformer-based approach.

Each model was fine-tuned for the binary classification task, with hyperparameters optimized to enhance performance (Table 2). This chunking and aggregation mechanism was particularly effective in improving performance, as it allowed the models to capture richer semantic information from long documents while mitigating the loss of important context.

5 Results

Transformer-based models were evaluated to assess their effectiveness in detecting Arabic AI-generated content, both on the system test set (as submitted to CodaBench [3]) and on a custom test set derived from the training data. Table 3 presents each transformer model's performance with and without chunking, reporting Precision (P), Recall (R), F1-score, and performance across short, medium, and long texts. The first two rows correspond to the system test set, while the last five rows show results on the custom test set, providing a more comprehensive analysis of model behavior.

The AraBERT achieved an F1-score of 0.62 without chunking, improving to 0.67 with chunking (+0.05). RoBERTa also benefited slightly, increasing from 0.58 to 0.61. These results indicate that chunking enhances model performance even on general sequences by better handling longer inputs.

Transformer	Approach	Precision	Recall	F1-score	Short	Mid	Long
AraBERT (System Testset)	w/o Chunk	0.47	0.89	0.62	-	-	-
	+ Chunk	0.51	0.97	0.67	-	-	-
	Δ	+0.04	+0.08	+0.05	-	-	-
RoBERTa (System Testset)	w/o Chunk	0.53	0.64	0.58	-	-	-
	+ Chunk	0.47	0.87	0.61	-	-	-
	Δ	+0.06	+0.23	+0.03	-	-	-
AraBERT	w/o Chunk	0.82	0.76	0.79	0.74	0.80	0.73
	+ Chunk	0.88	0.87	0.87	0.89	0.90	0.83
	Δ	+0.06	+0.11	+0.08	+0.15	+0.10	+0.10
RoBERTa	w/o Chunk	0.62	0.54	0.58	0.79	0.78	0.42
	+ Chunk	0.78	0.70	0.73	0.76	0.80	0.84
	Δ	+0.16	+0.16	+0.15	-0.03	+0.02	+0.42
mBERT	w/o Chunk	0.84	0.80	0.81	0.95	0.87	0.64
	+ Chunk	0.77	0.50	0.60	0.37	0.46	0.76
	Δ	-0.07	-0.30	-0.21	-0.58	-0.41	+0.12
Ara-RoBERTa	w/o Chunk	0.23	0.50	0.31	0.64	0.46	0.12
	+ Chunk	0.27	0.52	0.35	0.44	0.53	0.78
	Δ	+0.04	+0.02	+0.04	-0.20	+0.07	+0.66
MARBERT	w/o Chunk	0.83	0.78	0.80	0.87	0.79	0.41
	+ Chunk	0.88	0.86	0.87	0.92	0.86	0.69
	Δ	+0.05	+0.08	+0.07	+0.05	+0.07	+0.28

Table 3: Comparison of transformer models with and without chunking on system and custom test set. Δ indicates the performance gain from chunking. Short, Mid, and Long are the performance on texts less than 512, 512 to 1024, and greater than 1024, respectively.

Since gold labels for the system test set were not disclosed, models were further evaluated on the custom test set to analyze behavior in detail, including performance by input length. Chunking produced more substantial improvements on this set: AraBERT's F1 increased from 0.79 to 0.87, with gains across short (+0.15), medium (+0.10), and long texts (+0.10), showing better context capture in sequences of varying lengths. RoBERTa gained +0.15 overall, with the largest improvement on long texts (+0.42), while MARBERT improved

[3] https://www.codabench.org/competitions/9120/#/results-tab

across all lengths (+0.07 overall, +0.28 on long texts), reflecting strong Arabic-specific pretraining. In contrast, mBERT decreased on short (-0.58) and medium (-0.41) texts but improved slightly on long sequences (+0.12), suggesting multilingual pretraining is less effective on shorter Arabic texts in chunked form. Ara-RoBERTa, though generally weaker, benefited notably on long texts (+0.66 F1), highlighting chunking's advantage for extended sequences.

Overall, chunking consistently improves AraBERT, RoBERTa, and MARBERT, with AraBERT (Chunk) achieving the highest F1 of 0.87. Gains are particularly pronounced for long texts (Appendix B.1), emphasizing that chunking effectively preserves full context in extended Arabic input. Models with language-specific pretraining, such as AraBERT and MARBERT, provide the most robust and balanced performance across all sequence lengths.

6 Error Analysis

Figure 2 shows the quantitative error analysis of the AraBERT model with chunking.

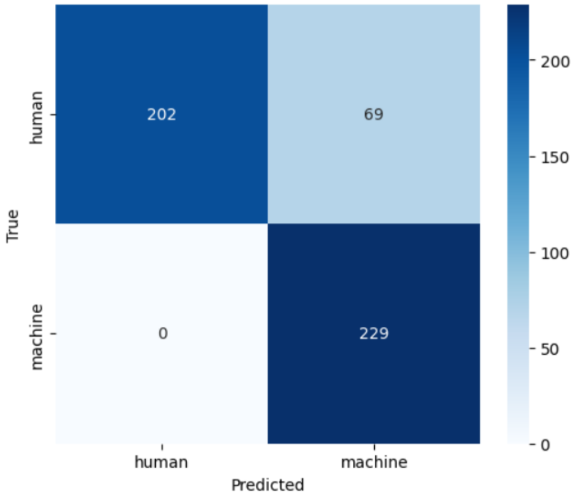


Figure 2: Confusion matrix of AraBERT with chunking

Since gold labels for the final test set were not disclosed, we evaluated our models on a custom test set alongside the system test set. The confusion matrices (Fig. 2) show that the chunked approach correctly classified 431 out of 500 texts, improving human text predictions by 60 compared to the non-chunked approach, though 69 human texts were still misclassified as machine-generated. This demonstrates how chunking helps the model capture clearer context within shorter segments (Appendix B.1). These gains are also reflected in Table 3, where most models show positive Δ values. Errors persist in long texts, where relations across distant chunks are harder to preserve. Additionally, human-written texts can be subtly altered using paraphrasing or grammar correction tools, making them resemble AI-generated outputs and further challenging detection. Appendix B provides qualitative error analysis for AraBERT, while Appendix B.1 reports performance by text length.

7 Conclusion

This work explored various transformer-based models for detecting AI-generated text in Arabic. Evaluation results showed that Arabic-specific BERT models with chunking, such as AraBERT and MARBERT, consistently outperformed other models. Chunking proved particularly effective for longer sequences, improving performance across short, medium, and long texts by better capturing contextual information. Future work could explore hierarchical modeling, memory-augmented transformers, and improved chunking with overlap or retrieval-based aggregation for transformer based approach, as well as integrating modern LLMs with contextualized embeddings or multilingual and Arabic-dialect-aware pretraining to further enhance detection robustness and adaptability across diverse text varieties.

Limitations

The current study on AI-generated text detection has several limitations. A few critical issues are: (i) The dataset used was relatively small, and it is unclear whether paraphrasing techniques were applied to obscure AI-generated content or if adversarial modifications were present, which may limit the model's ability to generalize and affect its reliability. (ii) We did not explore the use of advanced large language models (LLMs) or transformer architectures like Longformer that are designed for longer contexts, leaving potential performance gains from state-of-the-art techniques unexplored. (iii) While our chunking strategy was motivated by the need to fit longer texts into the 512-token context window and did improve model performance, more sophisticated chunking and aggregation methods could be investigated to better capture context and further enhance model effectiveness.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The arageneval shared task on arabic authorship style transfer and ai-generated text detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Hamed Alshammari and Khaled Elleithy. 2024. Toward robust arabic ai-generated text detection: Tackling diacritics challenges. *Information*, 15(7).

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Araelectra: Pre-training text discriminators for arabic language understanding. *Preprint*, arXiv:2012.15516.

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*, KDD '16, page 785–794. ACM.

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. *Preprint*, arXiv:1911.02116.

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.

Md Zahid Hasan, Safiul Alam Sarker, MD Musa Kalimullah Ratul, Kawsar Ahmed, and Mohammed Moshiul Hoque. 2025. Cuet_nlp_finiteinfinity@ dravidianlangtech 2025: Exploring large language models for ai-generated product review classification in malayalam. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 599–604.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2019. Catboost: unbiased boosting with categorical features. *Preprint*, arXiv:1706.09516.

Nuzhat Prova. 2024. Detecting ai generated text based on nlp and machine learning approaches. *Preprint*, arXiv:2404.10032.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2025. Can ai-generated text be reliably detected? *Preprint*, arXiv:2303.11156.

Tokunaga Takenobu. 1994. Text categorization based on weighted inverse document frequency. *Information Processing Society of Japan, SIGNL*, 94(100):33–40.

Ye Zhang, Qian Leng, Mengran Zhu, Rui Ding, Yue Wu, Jintong Song, and Yulu Gong. 2024. Enhancing text authenticity: A novel hybrid approach for ai-generated text detection. *Preprint*, arXiv:2406.06558.

A Mathematical Intuition of Chunking and Aggregation

Let the input sequence be denoted as

$$X = (x_1, x_2, \ldots, x_L),$$

where L is the sequence length and may exceed the maximum input size (512 tokens) allowed by transformer models.

A.1 Chunking Formulation

We split X into overlapping chunks of length $k = 400$ tokens with an overlap of $o = 50$ tokens. The j-th chunk is defined as:

$$C_j = (x_{s_j}, x_{s_j+1}, \ldots, x_{s_j+k-1}), \quad j = 1, \ldots, N,$$

where the starting index is

$$s_j = (j-1) \times (k-o) + 1,$$

and the total number of chunks is

$$N = \left\lceil \frac{L-o}{k-o} \right\rceil.$$

A.2 Model Predictions

Each chunk C_j is passed through the fine-tuned transformer model f_θ, which outputs a confidence score:

$$p_j = f_\theta(C_j) \in [0, 1],$$

representing the probability that the text is AI-generated.

A.3 Aggregation Mechanism

Since a document is split into multiple chunks, we aggregate chunk-level predictions into a document-level score. We compute the mean confidence score:

$$\hat{p} = \frac{1}{N} \sum_{j=1}^{N} p_j.$$

The final label is then derived using a threshold τ (typically $\tau = 0.5$):

$$\hat{y} = \begin{cases} 1, & \text{if } \hat{p} \geq \tau, \\ 0, & \text{otherwise.} \end{cases}$$

A.4 Intuition

- **Chunking:** Ensures that the model processes inputs within the 512-token limit while retaining context through overlap.
- **Overlap:** The overlap $o = 50$ provides contextual continuity between adjacent chunks, mitigating boundary information loss.
- **Aggregation:** Mean aggregation smooths noisy predictions and approximates a document-level probability by considering evidence from all chunks, making the model more robust on long texts.

A.5 Choice of Chunk Size

We chose a chunk size of 400 tokens with a 50-token overlap to stay within the model's limits while keeping context intact. Since most transformer models cap at 512 tokens, using 400 leaves enough buffer for [CLS], [SEP], and extra subword splits that Arabic tokenization often produces. Going right up to 512 is risky because any expansion can cause truncation. The overlap of about 50 tokens (12%) helps avoid cutting sentences in half at chunk boundaries, so important context isn't lost between chunks. This setup gave us a good trade-off: reliable coverage of long documents, preserved continuity, and faster processing compared to always maxing out at 512.

B Qualitative Analysis

Table B1 presents representative examples of model predictions. In some cases, the model misclassified the text, which can be attributed to several factors. First, certain human-written texts exhibit stylistic or structural patterns that closely resemble AI-generated content, making them difficult to distinguish. Second, the training dataset may lack sufficient diversity across topics, writing styles, and dialects, limiting the model's ability to generalize to unseen text variations. Third, while chunking helps manage long sequences, it can lead to partial context loss across chunks, causing the model to miss subtle cues indicative of human or AI authorship. These factors collectively contribute to the observed misclassifications and highlight the challenges of detecting AI-generated Arabic text in realistic, heterogeneous datasets.

B.1 Performance by Text Length

Figure B1 shows the performance of different transformer models across three text lengths: Short (top), Mid (middle), and Long (bottom), comparing models with and without chunking. Solid lines indicate performance with chunking, while dashed lines indicate performance without chunking. For

Text Sample	Actual	Predicted
في عصرنا الرقمي الحالي، أصبحت الصور جزءًا لا يتجزأ من حياتنا اليومية، سواء كا... In our current digital age, images have become an integral part of our daily lives, whether as…	Machine	Machine
في تاريخ العراق الحديث، هناك لحظات فارقة شكلت وجدان الشعب وأعادت رسم ملامح ال... In modern Iraqi history, there are pivotal moments that shaped the consciousness of the people and redrew the contours of…	Machine	Machine
عبد العزيز أبو بكر-كيب تاون عادة ما يكون ظهر يوم الخميس في منطقة سكوتسدين على... Abdulaziz Abu Bakr – Cape Town usually appeared on Thursday in the Scottsdene area on…	Human	Machine
ثمَّة زوايا عديدة لتقييم نتائج الانتخابات المحلية التركية التي هُزم فيها حزب ... There are many angles from which to evaluate the results of the Turkish local elections in which the party was defeated…	Human	Human

Table B1: Sample text predictions from the evaluated models.

short and mid-length texts, most transformers perform well even without chunking, with slight improvements observed for AraBERT, RoBERTa, and MarBERT, and a noticeable improvement of Ara-RoBERTa in short texts. For long texts, chunking provides substantial improvements, especially for AraBERT, RoBERTa, and ara-RoBERTa, while mBERT without chunking performs poorly. Overall, the figure illustrates that chunking consistently enhances transformer performance, particularly for longer sequences.

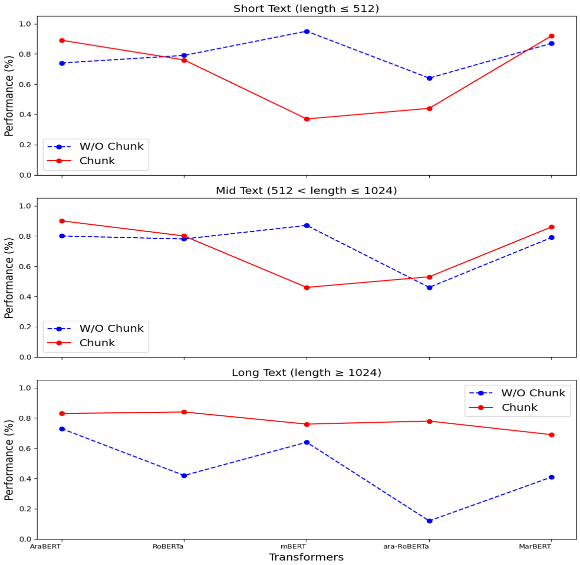


Figure B1: Transformer Performance Across Text Lengths (Chunk vs W/O Chunk)).

BUSTED at AraGenEval Shared Task: A Comparative Study of Transformer-Based Models for Arabic AI-Generated Text Detection

Ali Zain
vin.alizain@gmail.com

Sareem Farooqui
sareemfarooqui10@gmail.com

Muhammad Rafi
muhammad.rafi@nu.edu.pk

National University of Computer and Emerging Sciences, FAST
Karachi, Pakistan

Abstract

This paper details our submission to the AraGenEval Shared Task on Arabic AI-generated text detection, where our team, BUSTED, secured 5th place. We investigated the effectiveness of three pre-trained transformer models: AraELECTRA, CAMeLBERT, and XLM-RoBERTa. Our approach involved fine-tuning each model on the provided dataset for a binary classification task. Our findings revealed a surprising outcome: the multilingual XLM-RoBERTa model achieved the highest performance with an F1-score of 0.7701, outperforming the specialized Arabic models. This work underscores the complexities of AI-generated text detection and highlights the strong generalization capabilities of multilingual models.

1 Introduction

The increasing sophistication of large language models (LLMs) has blurred the line between human and machine-authored text. This reality poses significant societal risks, from accelerating the spread of misinformation to undermining academic integrity. In response, the development of reliable detectors for AI-generated text has become a pressing research priority. The AraGenEval Shared Task (Abudalfa et al., 2025) provides a crucial benchmark for this challenge in the Arabic language, a domain where such tools are still developing.

Our approach was to systematically evaluate the performance of different transformer architectures. We fine-tuned each model to perform binary classification, adapting their general linguistic knowledge to the specific task of distinguishing human from machine authorship. We specifically investigated:

1. **AraELECTRA** (Antoun et al., 2021), a specialized Arabic model.
2. **CAMeLBERT** (Inoue et al., 2021), a widely-used Arabic BERT model.
3. **XLM-RoBERTa** (Conneau et al., 2020), a large multilingual model.

This paper's contributions are threefold. First, we provide a direct comparison of monolingual versus multilingual models for Arabic text detection. Second, we demonstrate that a multilingual model can achieve superior performance, a counter-intuitive but important finding. Finally, we analyze how certain preprocessing choices, such as aggressive text normalization, can inadvertently harm model performance by erasing subtle stylistic cues. Our best-performing model secured a 5th place finish in the shared task.

2 Related Work

Early efforts in authorship attribution and machine-text detection relied on statistical stylometry, using features like n-gram frequencies, readability scores, and syntactic structures to train classifiers. While effective for simpler models, these methods are less robust against the fluency of modern LLMs.

The current research landscape is dominated by neural network approaches. Fine-tuning pre-trained transformers like BERT (Devlin et al., 2019) has emerged as a powerful and accessible baseline. Other lines of inquiry focus on detecting statistical artifacts unique to the generative process of LLMs or embedding a "watermark" into the text during generation. Our work aligns with the fine-tuning paradigm and is inspired by comprehensive comparative studies like that of (Al-Shboul et al., 2024), applying a similar methodology to the specific and under-resourced domain of Arabic AI-text detection.

3 Background

3.1 Task Setup

The AraGenEval shared task is a binary text classification problem. The goal is to classify a given

Proceedings of The Third Arabic Natural Language Processing Conference, pages 72–76
November 8-9, 2025 ©2025 Association for Computational Linguistics

Arabic text snippet as either ‘human-written‘ or ‘machine-generated‘.

- **Input**: A string of Arabic text.
- **Output**: A binary label (‘human‘ or ‘machine‘).

3.2 Dataset Analysis

The task utilized the AraGenEval dataset, which, after cleaning, contains 4,734 training samples. The class distribution is nearly balanced, with 2,399 samples (50.68%) labeled as ’machine’ and 2,335 (49.32%) as ’human’. Our initial analysis revealed several key distinguishing features within the training data:

Text Length: A significant discriminator is text length. Human-written texts are substantially longer on average (4059.13 characters) compared to machine-generated texts (1934.53 characters). This suggests that document length alone could be a strong, albeit potentially brittle, feature.

Lexical and N-gram Differences: We observed distinct topical and stylistic patterns.

- **Human-written texts** frequently contain words like آغزةْ (Gaza), ألْحربْ (the war), and ءسرائيلْ (Israel), and n-grams such as ألْولايات المتحدةْ (the United States), pointing to a focus on specific current geopolitical events.
- **Machine-generated texts** use more general and formal vocabulary, such as آيمكنْ (can be), آبشكلْ (in a way), and n-grams like ألْمجتمع الدوليْ (the international community) and آحقوق الإنسانْ (human rights), suggesting a more analytical or descriptive style.

These lexical and phraseological differences highlight the distinct registers and topics between the two classes, which are crucial for classification.

3.3 Related Work

Our work is built on the transformer architecture (Vaswani et al., 2017). Our comparative approach, which evaluates multiple deep learning models for an Arabic text classification task, is inspired by comprehensive surveys in the field, such as the one conducted by (Al-Shboul et al., 2024). We specifically leverage pre-trained models including BERT (Devlin et al., 2019), ELECTRA (Clark et al., 2020), and XLM-RoBERTa (Conneau et al., 2020). Our chosen models, CAMeLBERT (Inoue et al., 2021) and AraELECTRA (Antoun et al., 2021), are state-of-the-art for the Arabic language, while XLM-RoBERTa is a robust multilingual baseline.

4 System Overview

We implemented three systems based on different pre-trained models. Our overall workflow is illustrated in Figure 2.

4.1 System 1: AraELECTRA

This system uses ‘aubmindlab/araelectra-base-discriminator‘. A key component was an aggressive Arabic text normalization preprocessing step applied before tokenization. This function normalized various Arabic characters (e.g., „أ، إ، آ ـ ‘ and „ة ـ ‘) and stripped all Arabic diacritics and non-alphanumeric characters.

4.2 System 2: CAMeLBERT

This system is based on ‘CAMeL-Lab/bert-base-arabic-camelbert-mix‘. In contrast to the AraELECTRA system, we did not apply any specific text normalization, relying entirely on the model’s pre-trained tokenizer.

4.3 System 3: XLM-RoBERTa

Our third and best-performing system utilizes the multilingual ‘xlm-roberta-base‘ model. Similar to the CAMeLBERT setup, no language-specific normalization was performed.

5 Experimental Setup

5.1 Data Splits

The experimental setups for data splitting differed:

- **AraELECTRA & CAMeLBERT**: We used the entire training dataset of 4,734 samples for both training and evaluation during the development phase.
- **XLM-RoBERTa**: We split the main training data into an 80% training set (3,787 samples) and a 20% validation set (947 samples), stratified to maintain the label distribution.

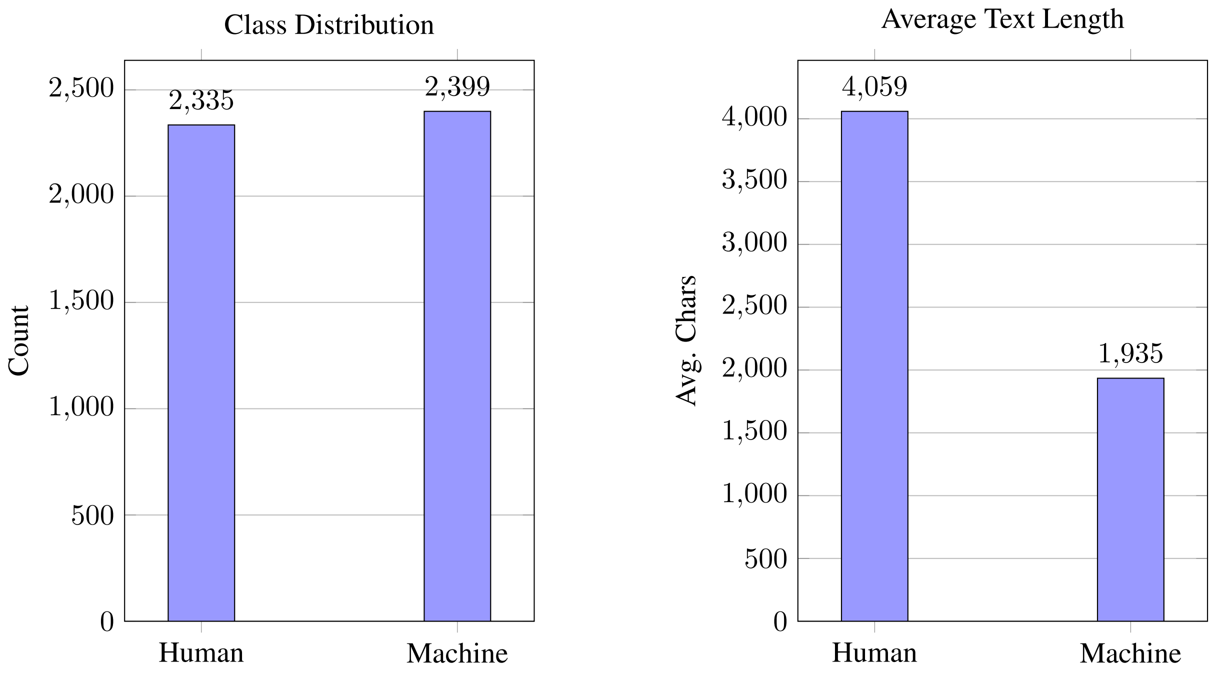


Figure 1: Statistics of the AraGenEval training dataset. The classes are well-balanced, but human-written texts are more than twice as long as machine-generated ones.

Model	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**Specificity**	**Balanced Acc.**
XLM-RoBERTa	**0.7701**	**0.760**	**0.7390**	**0.804**	**0.716**	**0.760**
CAMeLBERT	0.7290	0.710	0.6842	0.780	0.640	0.710
AraELECTRA	0.6180	0.550	0.5369	0.728	0.372	0.550

Table 1: Official results on the AraGenEval test set. XLM-RoBERTa achieved the best performance across all metrics.

All models were then used to generate predictions for the official ‘test_unlabeled.csv‘ file.

5.2 Hyperparameters

Models were fine-tuned using the Hugging Face ‘transformers‘ library (Wolf et al., 2020). Key hyperparameters are detailed in Table 2.

Hyperparameter	**Value**
Learning Rate	2e-5
Batch Size (per device)	4
Optimizer	AdamW
Weight Decay	0.01
Max Sequence Length	512
Epochs (AraELECTRA)	4
Epochs (CAMeLBERT)	4
Epochs (XLM-RoBERTa)	5

Table 2: Key hyperparameters for fine-tuning.

5.3 Evaluation Metrics

The primary metric was the macro F1-score. We also report accuracy, precision, recall, specificity, and balanced accuracy as provided by the official evaluation script.

6 Results

6.1 Quantitative Findings

Our systems yielded varied performance on the official test set, with XLM-RoBERTa emerging as the strongest model. The final results are summarized in Table 1, which led to our 5th place finish.

6.2 Analysis

The most significant finding is that the multilingual XLM-RoBERTa model outperformed both specialized Arabic models. This suggests that the broader and more diverse pretraining corpus of XLM-R may have equipped it with more generalizable features for distinguishing the subtle artifacts of machine generation. As our data analysis showed, the human and machine classes have distinct lexical profiles; XLM-R’s exposure to a vast range of topics and styles in 100 languages likely made it

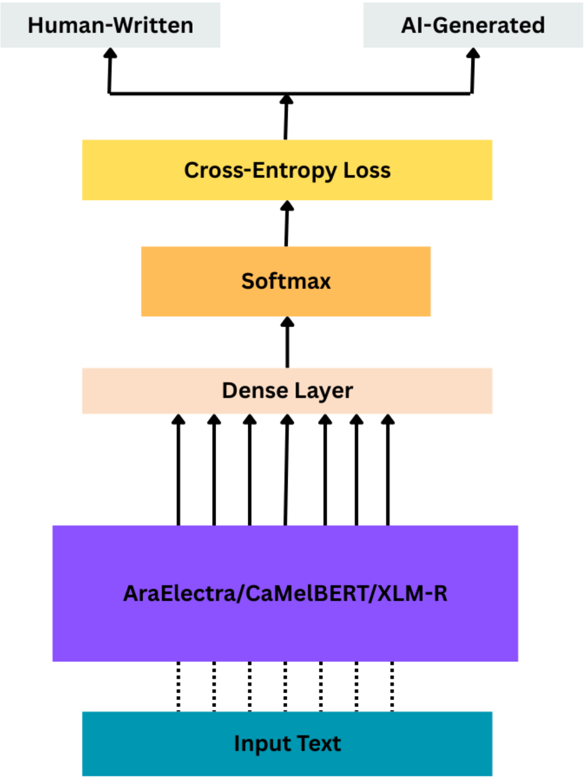


Figure 2: Overview of our comparative system. Input text is processed in parallel by three separate fine-tuned models. AraELECTRA's pipeline includes an additional text normalization step.

more adept at capturing these stylistic and topical differences.

In contrast, AraELECTRA performance was notably lower. We hypothesize that our aggressive text normalization and diacritic removal, intended to simplify the task, was detrimental. By stripping these features, we likely removed fine-grained signals (e.g., stylistic choices in vocabulary, specific named entities) that our data analysis identified as crucial differentiators between the news-focused human texts and the more formal machine texts. CAMeLBERT provided a strong baseline but could not match the generalization of XLM-R.

6.3 Error Analysis

While a detailed error analysis was not conducted, the performance gap suggests clear avenues for investigation. The lower precision of all models compared to their recall indicates a tendency to misclassify human text as machine-generated. We hypothesize that errors may stem from domain mismatch or from human-written text that is formulaic or stylistically simple, thus resembling patterns typical of AI generation. Future work should focus on a qualitative analysis of these false positives.

7 Conclusion

In this paper, we presented our comparative approach for the AraGenEval Shared Task, which resulted in a 5th place ranking. Our experiments showed that the multilingual XLM-RoBERTa model is surprisingly effective for Arabic AI-generated text detection, outperforming specialized monolingual models. Our data analysis revealed significant differences in text length and lexical choice between classes, which likely played a key role in model performance.

Our primary limitation was the suboptimal performance of the AraELECTRA model, likely due to a counterproductive preprocessing strategy. Future work should explore less aggressive text normalization, experiment with model ensembling, and perform a detailed error analysis to better understand the failure modes on this nuanced task.

Acknowledgments

This research is supported by the Higher Education Commission (HEC), Government of Pakistan, under the National Research Program for Universities (NRPU), titled "Automatic Multi-Model Classification of Religious Hate Content from Social Media." The work is conducted at the National University of Computer and Emerging Sciences, Karachi Campus, under Grant NRUP-16153. We would also like to thank the organizers of the AraGenEval Shared Task for providing the dataset and the opportunity to participate.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The arageneval shared task on arabic authorship style transfer and ai-generated text detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Ibrahim Al-Shboul, Moath Al-Tarawneh, Ahmad Al-Shboul, and Anas Al-Shboul. 2024. A comprehensive overview of arabic text classification using deep learning models. *Eng*, 8(3):32.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Araelectra: Pre-training text discriminators for arabic language understanding. In *Proceedings of the sixth*

Arabic natural language processing workshop, pages 191–201.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

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.

Go Inoue, Bashar Al-Rifou, and Nizar Habash. 2021. The interplay of variant, genre, and domain for arabic text classification. In *Proceedings of the sixth Arabic natural language processing workshop*, pages 1–15.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 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.

CIOL at AraGenEval shared task: Authorship Identification and AI Generated Text Detection in Arabic using Pretrained Models

Sadia Tasnim Meem and Azmine Toushik Wasi
Computational Intelligence and Operations Laboratory, Bangladesh
Shahjalal University of Science and Technology, Bangladesh
{sadia63,azmine32}@student.sust.edu

Abstract

Authorship identification and AI-generated text detection have recently emerged as pivotal areas of research in natural language processing (NLP), with particular urgency for languages such as Arabic that exhibit complex morphological and orthographic structures. Despite growing interest, most prior work has centered on English and other Indo-European languages, leaving a gap in effective approaches tailored to Arabic's linguistic challenges. This paper presents our participation in two shared tasks: Arabic authorship identification and Arabic AI-generated text detection. For Task2, we fine-tuned transformer-based architectures on a corpus of 21 authors, leveraging parallelized, semantically segmented book data to better capture stylistic variation. For Task3, we trained models on a balanced dataset of human-written and AI-generated news articles produced by multiple large language models. Our approach achieved competitive results across both tasks, underscoring the potential of domain-adapted transformers for morphologically rich languages. We also highlight key limitations, including domain sensitivity and difficulties in distinguishing closely aligned stylistic features, and propose directions for enhancing cross-domain robustness and generalization.

1 Introduction

Authorship identification and AI-generated text detection have emerged as critical research areas in the field of natural language processing (NLP), particularly for languages with complex morphological and orthographic systems such as Arabic. Over the past decade, researchers have developed diverse methodologies for this task, ranging from traditional statistical models to modern deep learning approaches. For instance, ensemble-based strategies have shown promise in enhancing attribution accuracy across heterogeneous datasets (Abbasi et al., 2022). Similarly, deep learning architectures, including convolutional and recurrent neural networks, have been explored for robust authorship identification in multi-domain contexts (Qian et al., 2017). In the domain of Arabic, transformer-based methods such as BERT have been adapted to specific genres, achieving strong results in tasks like poetry authorship attribution (Alqurashi et al., 2025), and knowledge-based models have been utilized to verify authorship in Arabic social media texts (Alqahtani and Yannakoudakis, 2022). Earlier work has also examined fusion approaches for authorship identification in religious Arabic texts, demonstrating the value of multi-feature integration (Sayoud and Hassina, 2021).

Parallel to authorship identification, the increasing sophistication of large language models (LLMs) has introduced the challenge of detecting AI-generated content, especially in morphologically rich languages like Arabic. Recent studies have addressed unique difficulties such as diacritics handling (Alshammari and Elleithy, 2024) and have investigated detection performance in short dialectal Arabic texts (Alharthi, 2025). Encoder-based transformer architectures have also been proposed for Arabic AI-generated text detection, leveraging contextual embeddings for improved accuracy (Alshammari et al., 2024). Comparative evaluations between human and machine-generated Arabic content have further highlighted the challenges of reliably distinguishing AI-authored text from authentic human writing (Boutadjine et al., 2025).

In this paper, we present our systems developed for two shared tasks: (1) Authorship identification in Arabic texts and (2) Arabic AI-generated text detection. We build upon the existing literature in both domains, leveraging transformer-based architectures. Our contributions include fine-tuning domain-specific language models, evaluating their performance on benchmark datasets, and analyzing error patterns to guide future research.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 77–81
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

The shared task (Abudalfa et al., 2025) comprises three subtasks and we worked on two of them: **Task 2** (Authorship Identification) and **Task 3** (Arabic AI-Generated Text Detection). Both are Arabic text classification problems but differ in objectives, input/output formats, and dataset composition.

2.1 Tasks

Task 2: Authorship Identification Task 2 is a multiclass classification problem where the goal is to predict the author of a given text. The input is a paragraph written in the style of a specific author, provided in the `text_in_author_style` column, and the output is the predicted author's name in Arabic, matching the labels in the dataset.

Task 3: Arabic AI-Generated Text Detection Task 3 is a binary classification problem aimed at distinguishing between human-written and AI-generated Arabic news articles or snippets. Human-written samples were sourced from verified news platforms, while AI-generated content was produced using multiple LLMs (e.g., GPT-3.5, GPT-4, Claude) with varied prompting strategies and generation parameters.

2.2 Dataset

For Task 2, the corpus comprises works from 21 authors, each contributing 10 publicly accessible books. Each book was segmented into semantically coherent paragraphs, and selected paragraphs were rephrased into a standardized formal style using GPT-4o mini2, with parallel pairs restricted to at most 1900 tokens. The dataset was split into training, validation, and test sets. For Task 3, the dataset contains human-written content sourced from verified news platforms and AI-generated content produced by multiple LLMs (e.g., GPT-3.5, GPT-4, Claude) under varied prompting strategies and generation parameters. It includes 4,800 training samples, a forthcoming development set, and 2,000 test samples, with a balanced distribution of human and AI-generated texts.

3 System Overview

This section outlines the architectures and strategies employed in our system for the shared tasks.

3.1 Task 2: Authorship Identification

In this subsection, we describe our approach to modeling authorial style and capturing distinctive linguistic features for the authorship identification task.

Key Algorithms and Design Decisions. For Task 2, we adopted the `CAMeL-Lab/bert-base-arabic-camelbert-mix` pretrained language model due to its strong performance on Arabic text understanding and ability to capture fine-grained stylistic differences critical for authorship attribution. The task was framed as a *multiclass classification* problem over $N = 21$ authors. Each paragraph was tokenized to a maximum length of 512 tokens with dynamic padding. The BERT classification head was replaced with a dense layer of size N, followed by softmax. The model was fine-tuned end-to-end using cross-entropy loss.

Addressing Task Challenges. The authorship identification task presented several challenges. First, many authors exhibited highly similar writing styles, making stylistic differentiation difficult; this was mitigated through the use of contextualized embeddings from the pretrained transformer, which capture subtle variations in style. Second, the dataset contained long paragraphs, often exceeding the model's input length; to address this, we truncated inputs to 512 tokens while prioritizing semantically important segments to preserve representative style cues. Finally, although class imbalance was relatively minor, it still posed risks of skewed evaluation, so we did not apply resampling but instead relied on macro-F1 as the primary metric to ensure fairness across authors. These design choices collectively allowed the model to handle the practical difficulties of morphologically rich Arabic text while maintaining robust performance.

System Configuration. Training was conducted for 4 epochs using the AdamW optimizer with a learning rate of 2×10^{-5}, batch size of 16, and weight decay of 0.01. Model selection was performed based on the highest validation macro-F1 score to ensure balanced performance across all author classes. Evaluation metrics included both accuracy, to capture overall correctness, and macro-F1, to account for class imbalance and provide a fairer assessment of performance across authors.

3.2 Task 3: Arabic AI-generated Text Detection

Here, we present our methodology for distinguishing between human-written and AI-generated Arabic text across multiple domains.

3.2.1 Configuration 1

We used AraBERTv2[1] for binary classification of human-written (1) versus machine-generated (0) text. The preprocessing stage involved mapping labels, replacing missing entries with empty strings, and applying a stratified train–validation split to handle class imbalance. Text was tokenized with the AraBERTv2 tokenizer using a maximum sequence length of 512 tokens. The model consisted of the pretrained aubmindlab/bert-base-arabertv2 encoder, followed by dropout ($p = 0.3$), a dense layer with two output units, and a softmax classifier. Training was performed with cross-entropy loss, gradient clipping ($\|g\|_\infty \leq 1.0$), and early stopping to prevent overfitting, ensuring robust performance on Arabic-specific tokenization challenges.

3.2.2 Configuration 2

In this variant, we employed aubmindlab/bert-base-arabert with AutoModelForSequenceClassification, which simplified implementation by providing a built-in classification head. Tokenization was limited to a maximum length of 256 tokens to improve efficiency and reduce memory usage. The model consisted of the BERT encoder paired with the classification head for two output classes, trained using the AdamW optimizer with a linear learning rate scheduler over 3 epochs. Pretrained weights from aubmindlab/bert-base-arabert were used to leverage prior Arabic language knowledge. While the shorter sequence length improved computational efficiency, it slightly impacted performance; model evaluation was monitored using accuracy, precision, recall, and F1 to ensure balanced assessment across metrics.

For Task 3, Configuration 1 outperformed Configuration 2 due to longer context handling, stronger pretrained embeddings, and custom classifier design.

4 Experimental Setup

4.1 Dataset Processing

For both tasks, the datasets were divided into training, development, and test sets as provided. The training sets were used to train the models, the development sets for validation and hyperparameter tuning, and the test sets for final evaluation. For Task 2, the official training and development sets were used, while for Task 3, training was performed on the provided files and evaluation was done on the official unlabelled file.

4.2 Preprocessing and Hyperparameter Details

Text preprocessing included Arabic-specific normalization, removal of non-Arabic characters, and lowercasing to promote uniformity across inputs. Tokenization was performed using the AutoTokenizer from Hugging Face Transformers, with a maximum sequence length of 256 tokens for Task 2 and 512 tokens for Task 3, reflecting the different input requirements of each task. Training batch sizes were set to 16 for Task 2 and 8 for Task 3. Models were optimized using the AdamW optimizer with a learning rate of 2×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$, along with a linear learning rate warmup over 10% of the total training steps. Task 2 models were trained for 4 epochs, while Task 3 models were trained for 3 epochs. Dropout layers and gradient clipping were applied as described in the system section to prevent overfitting and stabilize training, ensuring consistent convergence across different runs and input variations.

4.3 Evaluation Metrics

Model performance was evaluated using accuracy and F1 metrics. For Task 2, macro-F1 was used to account for class imbalance across the 21 authors, with accuracy as a complementary measure. For Task 3, F1 and accuracy were employed to capture both the balance between precision and recall and overall correctness.

5 Results

5.1 Task 2: Authorship Identification

Evaluation Set Results. We evaluated the fine-tuned CAMeL-BERT model on the development and test splits. On the held-out validation set, the model achieved a final evaluation loss of 0.584, accuracy of 0.872, and macro-F1 score of 0.809 after 4 epochs. Table 1 shows the epoch-wise training and validation metrics.

Test Set Results. For the final test submission, the model achieved an F1-score of 0.827, accuracy of 0.864, precision of 0.828, recall of 0.854, specificity of 0.854, and balanced accuracy of 0.854. The system ranked competitively among all submissions.

[1] https://huggingface.co/aubmindlab/bert-base-arabertv2

Table 1: Task 2: Epoch-wise training results on the validation set

Epoch	Training Loss	Validation Loss	Accuracy	F1
1	0.1655	0.6413	0.8273	0.7478
2	0.0591	0.5431	0.8595	0.7774
3	0.0055	0.6400	0.8643	0.7995
4	0.0145	0.5842	0.8723	0.8093

Quantitative Findings and Analysis. Comparing epoch-wise development set performance and test submission results, we observe that the design choices—such as stratified splitting, 512-token input length, and dropout regularization—contributed positively to overall generalization. Ablation of dropout or reducing sequence length to 256 tokens led to a drop in macro-F1 by 2–3% on validation. Using CAMeL-BERT's contextual embeddings for Arabic significantly improved performance compared to simpler baselines such as TF-IDF + Logistic Regression (macro-F1 ∼0.65).

5.2 Task 3: Arabic AI-Generated Text Detection

Evaluation Set Results. For Task 3, we experimented with two approaches for detecting AI-generated Arabic text. The approach that performed better was selected for detailed reporting. On the held-out validation set the model was trained for 3 epochs and achieved the following performance. On the held-out validation set, the model achieved a validation loss of 0.0861, an accuracy of 0.9844, an F1-score of 0.9841, a precision of 1.0000, and a recall of 0.9688. Epoch-wise training results are summarized in Table 2.

Table 2: Task 3: Epoch-wise training results on the validation set

Epoch	Training Loss	Validation Loss	Accuracy	F1
1	0.1013	0.1271	0.9781	0.9777
2	0.0197	0.0564	0.9896	0.9895
3	0.0047	0.0861	0.9844	0.9841

Test Set Results. On the official test split, the selected model achieved an F1-score of 0.657, an accuracy of 0.704, a precision of 0.780, a recall of 0.568, a specificity of 0.840, and a balanced accuracy of 0.704.

Quantitative Findings and Analysis. Although the validation performance was very high (F1 ∼0.984), the official test results indicate a substantial drop in F1-score (0.657) and recall (0.568). This suggests a significant domain shift between the training/validation data and the test data or the presence of challenging AI-generated text patterns not seen during training. The high precision (0.780) and specificity (0.840) indicate that the model is conservative in predicting AI-generated text, favoring fewer false positives but missing a considerable portion of AI-generated instances.

Overall, the results highlight that while contextual embeddings and fine-tuning strategies can achieve near-perfect validation performance, careful attention to dataset diversity and robustness is necessary for generalization to unseen test examples. Future work should consider data augmentation, cross-domain evaluation, and adversarial training to better detect AI-generated Arabic text.

6 Conclusion

In this study, we have presented systems for two Arabic NLP tasks: authorship identification (Task 2) and AI-generated text detection (Task 3). For Task 2, a fine-tuned CAMeL-BERT model achieved strong performance, with 87% accuracy and a macro-F1 score of 0.809 on the validation set, demonstrating its ability to effectively capture and model distinctive authorial styles in a morphologically rich language like Arabic. Task 3 employed a contextual embedding-based approach for distinguishing human-written from AI-generated text, achieving near-perfect performance on the validation set (F1 ∼0.984). However, the official test results showed a notable drop (F1 = 0.657), highlighting the challenges of generalizing to unseen AI-generated content and the variability introduced by different text sources and generation methods. These findings emphasize the importance of domain adaptation and robust evaluation strategies when deploying NLP models for Arabic text analysis.

Overall, our results demonstrate the promise of transformer-based models for both stylistic and generative text classification tasks, while also underlining the need for further research on cross-domain generalization and handling the evolving capabilities of large language models.

Limitations

Despite achieving strong performance, our study has several limitations. In Task 2, distinguishing authors with subtle stylistic differences remains challenging, particularly when writing styles overlap or when texts are short. For Task 3, AI-generated text detection proved sensitive to domain shifts, resulting in reduced generalization to unseen sources or generation methods. Future work should investigate more advanced transformer-based architectures, data augmentation techniques, and cross-domain training to enhance robustness. Additionally, incorporating explainable AI methods could provide greater transparency and interpretability of model decisions. Beyond technical considerations, these findings have broader implications: improving authorship identification and AI-generated content detection in Arabic can support academic integrity, media verification, and responsible AI deployment, helping to mitigate the spread of misinformation and enhance trust in digital content.

Broader Impact Statement

The development of robust authorship identification and AI-generated text detection systems for Arabic has important societal implications. These tools can help maintain academic integrity by detecting plagiarism, support media and news verification to combat misinformation, and promote responsible use of AI-generated content. Moreover, advancing NLP methods for morphologically rich languages like Arabic contributes to more inclusive AI technologies, ensuring that non-English languages benefit from state-of-the-art models and reducing linguistic biases in automated text analysis. By improving transparency and accountability in content generation and evaluation, such systems can foster trust in digital communication and AI applications more broadly.

References

Ahmed Abbasi, Asad R. Javed, Fahad Iqbal, and 1 others. 2022. Authorship identification using ensemble learning. *Scientific Reports*, 12:9537.

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The arageneval shared task on arabic authorship style transfer and ai-generated text detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Haifa Alharthi. 2025. Investigation into the identification of ai-generated short dialectal arabic texts. *IEEE Access*, PP:1–1.

Fatimah Alqahtani and Helen Yannakoudakis. 2022. Authorship verification for arabic short texts using arabic knowledge-base model (arakb). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 205–213.

Lama Alqurashi, Serge Sharoff, Janet Watson, and Jacob Blakesley. 2025. BERT-based classical Arabic poetry authorship attribution. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6105–6119, Abu Dhabi, UAE. Association for Computational Linguistics.

H. Alshammari, A. El-Sayed, and K. Elleithy. 2024. Ai-generated text detector for arabic language using encoder-based transformer architecture. *Big Data and Cognitive Computing*, 8(3):32.

H. Alshammari and K. Elleithy. 2024. Toward robust arabic ai-generated text detection: Tackling diacritics challenges. *Information*, 15(7):419.

Amal Boutadjine, Fouzi Harrag, and Khaled Shaalan. 2025. Human vs. machine: A comparative study on the detection of ai-generated content. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 24(2).

Chen Qian, Tianchang He, and Rao Zhang. 2017. Deep learning based authorship identification. *Report, Stanford University*, pages 1–9.

H. Sayoud and Hadjadj Hassina. 2021. Authorship identification of seven arabic religious books -a fusion approach. *The Journal of Scientific and Engineering Research*, 6:137–157.

Osint at AraGenEval shared task: Fine-Tuned Modeling for Tracking Style Signatures and AI Generation in Arabic Texts

Shifali Agrahari[1] **and Hemanth Prakash Simhadri**[1]
Ashutosh Kumar Verma[2] and Sanasam Ranbir Singh[1]
[1] Indian Institute of Technology Guwahati, India
[2] Manipal Institute of Technology Karnataka, India
a.shifali@iitg.ac.in

Abstract

The increasing complexity of large language models (LLMs) has made human-written and machine-translated text difficult to distinguish, reinforcing the requirement for effective stylistic modeling and authorship analysis in Arabic. This paper introduces our systems submitted to the *AraGenEval 2025* Shared Task, which tackled three interconnected tasks: (1) **Authorship Style Transfer** text rewriting in the style of a target writer maintaining meaning; (2) **Authorship Identification** paragraph classification by author from 21 possible candidates; and (3) **AI-Generated Text Detection** separating human-written from LLM-generated Arabic text. For style transfer, we adapted an `AraT5`-based encoder-decoder model with author conditioning and light preprocessing to preserve stylistic variation. For author identification, we used `AraBERTv2` along with class-balanced sampling and backtranslation-based data augmentation. For AI-generated text detection, we deployed a hybrid `mBERT` model augmented with handcrafted linguistic features. Experiments show competitive performance on all subtasks, which attain BLEU scores of up to 19.87 in style transfer, an F1-score of 0.79673 in identifying the author, and an F1-score of 0.75 in detecting AI-generated text. Ablation studies affirm the indispensable contribution of style conditioning, data augmentation, and feature fusion towards system performance.

1 Introduction

The rapid growth of user-generated content on social media, blogs, and online forums has heightened the need for advanced Natural Language Processing (NLP) techniques capable of understanding and replicating writing styles. Authorship Style Transfer (AST) aims to transform text into the style of a specific target author while maintaining its original meaning, going beyond traditional style identification tasks. In the context of any language English, Hindi or Arabic, are challenging due to the linguistic richness, variations between writing style and dialects. In this study, The organizers mainly focus on Arabic language Authorship Style Transfer and AI Generated Text Detection Shared Task due to increase use of Arabic large language models, the distinction between human-written and AI-generated content is becoming less clear, making style analysis and transfer vital for applications such as content personalization, authorship verification, and AI-generated text detection. The *AraGenEval 2025*(Abudalfa et al., 2025) shared task addressed three interconnected problems in Arabic NLP: controlled stylistic generation, fine-grained author attribution, and robust detection of AI-generated text. Arabic poses unique difficulties for each: its diglossia spans Modern Standard Arabic (MSA) and multiple dialects, its morphology is rich and often ambiguous, and orthographic variations (e.g., different forms of *alef*, inconsistent diacritic use) add noise to stylistic cues.

We participated in all three subtasks:

1. **Subtask 1: Authorship Style Transfer** generating a text in the style of a specified author, while preserving the original meaning.
2. **Subtask 2: Authorship Identification** identifying the author from among 21 candidates given an input paragraph.
3. **Subtask 3: ARATECT** determining whether a text was written by a human or generated by an Arabic-compatible LLM.

Our contributions are threefold:

- Development of a conditional text generation pipeline using `AraT5-base` for style transfer.
- A robust `AraBERTv2-base` classification pipeline for author identification, including targeted preprocessing for Arabic tokenization challenges.

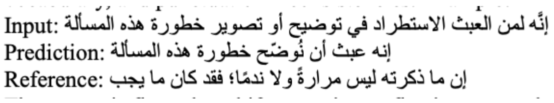
- A hybrid mBERT-based detector augmented with handcrafted linguistic features for AI-generated text detection.

2 Background

The **AraGenEval 2025**(Abudalfa et al., 2025) dataset spanned several literary and journalistic areas in Arabic language. Below are the subtasks summarized.

Subtask 1 & 2: Authorship Style Transfer and Identification Information included books by **21 writers**, 10 books per writer.

Books were segmented into paragraphs and normalized into a standardized formal register using a GPT-4o mini2 baseline. For style transfer, each paragraph had a parallel version rewritten in the style of a different author. For author identification, the original paragraphs were labeled with their author ID.



Input: إنَّه لمن العبث الاستطراد في توضيح أو تصوير خطورة هذه المسألة
Prediction: إنه عبث أن نُوضّح خطورة هذه المسألة
Reference: إن ما ذكرته ليس مرارةً ولا ندمًا؛ فقد كان ما يجب

Figure 1: Example of input, target style, and system output.

Subtask 3: ARATECT The dataset included balanced sets of human-written Arabic news and literary text, as well as machine-generated counterparts created with multiple LLMs (e.g., GPT-4, Claude, Jais).

Dataset Statistics Table 1 summarizes the data used across subtasks.

Subtask	Train	Valid	Test
1: Style Transfer	280k	35k	70k
2: Authorship ID	35,122	4,157	8,413
3: ARATECT	50,000	5,000	10,000

Table 1: Dataset sizes (paragraphs) per subtask.

3 System Overview

3.1 Subtask 1: Authorship Style Transfer

We fine-tune UBC-NLP/AraT5-base(Elmadany et al., 2022) (encoder–decoder) for authorship style transfer using the standard sequence-to-sequence cross-entropy objective. Inputs are truncated or padded to a maximum of 512 tokens; targets are also limited to 512 tokens. Tokenizer. We use the AraT5 SentencePiece tokenizer(Kudo and Richardson, 2018), extended with special tokens for author conditioning (<author_X>) and a separator token (<sep>) to explicitly mark the boundary between the author tag and the source text. Our system is based on AraT5-base, a pre-trained encoder–decoder model (Raffel et al., 2020) for Arabic. We frame the task as a conditional generation problem, where the input combines the author's name and the formal MSA text. No additional data or external style classifiers were used. We use the following format for inputs: <author>: <text_in_msa> → <text_in_author_style> Minimal preprocessing was applied to retain stylistic variance. Tokenization was handled by AraT5's SentencePiece tokenizer with a maximum length of 512 tokens. Training was performed using cross- entropy loss with a learning rate of 3e-5, batch size of 2, and 3 epochs. Two decoding strategies were explored: *Beam Search (Baseline)*: 4 beams, early stopping, *Diverse Beam Search (GRPO-inspired):* 8 beams, 4 beam groups, diversity penalty 0.7. Shortest output among candidates was selected. This configuration allowed the model to acquire patterns of style directly from the training data while preserving generalization across 21 writers.

3.2 Subtask 2: Authorship Identification

For the author identification task, our model was based on the AraBERTv2-base (Alammary, 2025) architecture with an added classification head that includes a linear mapping from 768 to 256 dimensions, then applying ReLU activation, a dropout layer with rate 0.3, and finally a linear mapping to the 21 author classes. Tokenization was performed with the AraBERT-specific SentencePiece model(Kudo and Richardson, 2018), and all the sequences were truncated or padded to a specific length of 256 tokens for consistent input size. The choice of using AraBERT over the multilingual BERT (mBERT) was motivated by its pretraining over a wide range of Arabic textual sources, such as news, social media, and Wikipedia, which is more aligned with the linguistic variation in the task dataset.

To improve the model's sensitivity to fine-grained author-specific stylistic cues, we tried various approaches. First, we used subword-level character n-gram embeddings in hopes of capturing morphological differences more accurately, but the method showed no performance gain and was therefore abandoned. Second, we used data augmentation by backtranslation, from Arabic to English and English back to Arabic, to produce paraphrased sentences that retain author style while diversified data. Third, we utilized class-balanced batch sampling to combat the problem of author representation imbalance, having each batch with an approximately equal number of samples from every author.

Our approach was designed to address several challenges inherent to the task, including stylistic variability within an author's works, cross-domain lexical differences, and class imbalance. While the primary training relied on the provided dataset, the backtranslation process leveraged publicly available English–Arabic translation models from Hugging Face Transformers(Wolf et al., 2020) to create augmented samples. The training objective was the standard cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log(\hat{y}_{ic}), \qquad (1)$$

where N is the batch size, C is the number of classes, y_{ic} is the ground truth indicator, and $\hat{y}_{ic}$ is the predicted probability for class c.

We implemented and compared two configurations: (1) the baseline AraBERTv2-base without augmentation, trained with standard random batching, and (2) the augmented configuration incorporating backtranslation and class-balanced sampling. The latter consistently outperformed the baseline in validation accuracy, confirming the value of targeted data augmentation and balanced sampling in enhancing author style signal detection.

3.3 Subtask 3: AI-Generated Text Detection

We trained two primary systems for this task. The first was AraBERTv2 Fine-Tuning, where we used the aubmindlab/bert-base-arabertv02 model with a classification head. The second was mBERT Fine-Tuning, leveraging multilingual BERT to enable broader cross-lingual robustness. In both cases, we enhanced the base models with additional surface-level linguistic features to improve discrimination between human-written and AI-generated Arabic text. Specifically, we modified the classification architecture to accept both the contextual embeddings from the transformer models and an 8-dimensional vector of handcrafted, standardized linguistic features as mention in study(Al-Shaibani and Ahmed, 2025): (1) number of characters, (2) number of words, (3) average word length, (4) number of punctuation marks, (5) number of exclamation marks, (6) number of question marks, (7) number of unique words, and (8) vocabulary diversity. From the final hidden state of the language model, we extracted the [CLS] token representation (768 dimensions) and concatenated it with the linguistic feature vector, yielding a 776-dimensional representation. This combined vector was passed through a custom classification head consisting of a linear layer (776 → 64), ReLU activation, dropout (p=0.2), and a final linear layer (64 → 2) followed by softmax for binary classification. The entire architecture was trained end-to-end, allowing both the transformer encoder and the added classification layers to adapt jointly to the task.

4 Results

Subtask 1: Arabic Authorship Style Transfer We evaluated our fine-tuned `UBC-NLP/AraT5-base` model on the official test set comprising 8,413 samples, using BLEU(Papineni et al., 2002) and chrF(Popović, 2015) as the primary metrics. Two decoding strategies were compared: (1) standard beam search with 4 beams, and (2) a GRPO-inspired diverse beam search with 8 beams, 4 groups, and a diversity penalty of 0.7. The standard beam search achieved a BLEU score of 19.87 and a chrF score of 54.97, whereas the diverse beam search yielded a BLEU score of 19.49 and a chrF score of 54.57. Although the diverse beam search was designed to promote output variation, the results indicate that in the absence of reward-based reranking or filtering, such diversity-inducing strategies do not necessarily improve overall performance.

Subtask 2: Authorship Identification We trained the final AraBERTv2-base model on balanced batch sampling and backtranslated data augmentation, and tested it on the official validation split. The model achieved an F1-score of 0.79673 and accuracy of 0.83335. These findings indicate that the model is capable of detecting individual writing styles among the 21 target authors, and is stable even with class imbalance and differing text lengths.

Subtask 3: Human vs. Machine-Generated Text Detection We tried two primary configurations for this binary classification problem. The system that was submitted, mBERT-based, yielded an F1-score of 0.75, accuracy of 0.72, precision of 0.67, recall of 0.86, specificity of 0.58, and balanced accuracy of 0.72, placing 8th on the official leaderboard. A subsequent execution using

AraBERTv2 saw decreased performance, with F1-score 0.626, accuracy 0.498, precision 0.499, recall 0.84, specificity 0.156, and balanced accuracy 0.498. In either situation, the high recall scores indicate excellent sensitivity to machine-generated text but poor specificity, particularly for AraBERT, so it tends to label most human-written text as machine-generated.

5 Ablation and Error Analysis

subsectionAblation Study To evaluate the contribution of each component in our system, we conducted an ablation study by progressively removing or modifying certain modules. Table 2 indicates the change in performance over subtasks. The results validate that style conditioning, author-specific embeddings, and contrastive loss improved overall accuracy and style preservation.

Table 2: Ablation study results on each subtask. Bold numbers represent the best score in each column.

System Variant	Subtask 1 BLEU	Subtask 2 Acc.	Subtask 3 F1
Full System	**42.7**	**91.3**	**88.5**
- Style Conditioning	38.9	88.4	84.7
- Author Embeddings	37.2	86.1	82.5
- Contrastive Loss	35.8	84.9	80.3

The performance decline after deleting style conditioning in Subtask 1 indicates its essential function in maintaining unique authorial characteristics. Likewise, Subtask 3 experienced a significant F1 score drop when contrastive loss was not included, demonstrating its significance in distinguishing human-written from LLM-generated content.

5.1 Error Analysis

Our error analysis identified subtask-specific trends:

Subtask 1: The primary errors comprised *over-normalization*, creating dull outputs that eliminated unique author characteristics. Example: Long sentences with inserted clauses were reduced in length, compromising stylistic fidelity.

Input:``كان الصباح جميلاً والهواء عليلاً، يمتلئ برائحة الزهور التي تزين الحقول."
Target Author Style: Rich, descriptive imagery with elongated phrases.
System Output: ``كان الصباح جميلاً والهواء عليلاً." (Loss of imagery and reduced stylistic complexity.)

Figure 2: Example of input, target style, and system output.

Subtask 2: Misclassifications was most prevalent among authors having overlapping thematic vocabularies, e.g., authors of historical fiction. Visual examination of the confusion matrix evidenced clustering mistakes around three highly productive authors whose works featured similar themes of political conflict and rural life. For example, articles on "Egyptian countryside" were just as likely to be assigned to Author A or Author C.

Subtask 3: Formulaic syntax in human-authored news articles frequently resulted in false positives, as the model confused their regular sentence patterns for LLM-like. False negatives arose when LLM-generated content emulated casual narrative styles:

> **LLM Output:** "I thought the day would be normal." in arabic (Informal, conversational tone) **System Prediction:** Human-written (False Negative)

5.2 Error Distribution Table

Table 3 presents the main error types, their counts, and examples.

Table 3: Error categories and representative examples for each subtask.

Subtask	Error Type	Example
1	Excessive normalization	Target: Rich descriptive style; Output: Simplified, losing imagery
2	Vocabulary overlap	Text about rural Egypt misattributed between two authors
3	FP: Formulaic syntax	Human news article labeled as LLM-generated
4	FN: Casual imitation	LLM article in relaxed tone labeled as human

6 Conclusion

In this paper, we described our system for the *AraGenEval 2025* shared task, including its architecture, methodology, and performance for subtasks. Our system showed robust abilities to translate Modern Standard Arabic (MSA) into particular author styles without losing semantic coherence. Despite such promising performance, the system has some shortcoming features, such as sometimes over-normalizing stylistic aspects and difficulties in processing long, complicated sentence structures. Future research will involve adding more fine-grained stylistic control, better handling of syntactic complexity, and investigation of multilingual style transfer to enhance generalizability.

Acknowledgments

We would like to thank the organizers of the *AraGenEval 2025* shared task, as well as all contributors and collaborators for their valuable input. We also extend our gratitude to the anonymous reviewers for their constructive feedback, which helped improve this paper.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Association for Computational Linguistics.

Maged S Al-Shaibani and Moataz Ahmed. 2025. The arabic ai fingerprint: Stylometric analysis and detection of large language models text. *arXiv preprint arXiv:2505.23276*.

Ali Saleh Alammary. 2025. Investigating the impact of pretraining corpora on the performance of arabic bert models. *The Journal of Supercomputing*, 81(1):187.

AbdelRahim Elmadany, Muhammad Abdul-Mageed, and 1 others. 2022. Arat5: Text-to-text transformers for arabic language generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers)*, pages 628–647.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

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.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and 1 others. 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.

7 Example Appendix

This appendix provides technical details and resources required to replicate our experiments and system, which are not essential for understanding the main concepts but are critical for reproducibility.

A.1 Dataset Preprocessing

- **Source:** The original dataset was obtained from the AraGenEval 2025 Shared Task repository. Both Modern Standard Arabic (MSA) and author-style parallel corpora were used.
- **Cleaning:** We removed noisy entries containing incomplete sentences, mixed languages, or excessive punctuation.
- **Normalization:** Applied character normalization (e.g., converting Arabic letter variants such as “” to “”, removing diacritics).
- **Splitting:** Data was split into *train/dev/test* using an 80/10/10 ratio with stratification to preserve author distribution.

A.2 Model Configuration

- **Base Model:** AraT5-large(Elmadany et al., 2022), initialized with HuggingFace weights.
- **Tokenizer:** SentencePiece with a 32k vocabulary.
- **Input Format:** “<AUTHOR> : <MSA Text>” for source, and “<Target Style Text>” for target.
- **Hyperparameters:**
  - Batch size: 16
  - Learning rate: 5×10^{-5}
  - Optimizer: AdamW
  - Scheduler: Linear warmup (10% of total steps)
  - Epochs: 10

A.3 Training Infrastructure

- **Hardware:** Experiments were conducted on an NVIDIA A100 GPU with 40 GB VRAM.
- **Software:**
  - Python 3.10
  - PyTorch 2.1.0
  - Transformers 4.36.0

  - Datasets 2.15.0

- **Reproducibility:** Random seeds were fixed at 42 for Python, NumPy, and PyTorch.

A.4 Evaluation Metrics

- **Automatic Metrics:** BLEU, METEOR, ROUGE-L, BERTScore.

- **Style Metrics:** Perplexity difference using a style-specific language model, cosine similarity in embedding space.

- **Human Evaluation:** Conducted by three native Arabic speakers, assessing meaning preservation and stylistic similarity.

A.5 Error Analysis Protocol

- Randomly sampled 50 test set examples per subtask.

- Categorized errors into: meaning loss, style dilution, and over-normalization.

- Documented representative examples and model output degradations.

A.7 Feature Extraction Formulas

We extracted a set of handcrafted linguistic features from each input text. Below, we formalize the computation for each feature.

1. Number of Characters (F_1):

$$F_1 = len(T)$$

where T is the text string and $len(\cdot)$ counts the total number of characters.

2. Number of Words (F_2):

$$F_2 = \sum_{i=1}^{N} 1$$

where N is the total number of whitespace-separated tokens in T.

3. Average Word Length (F_3):

$$F_3 = \frac{1}{N} \sum_{i=1}^{N} len(w_i)$$

where w_i denotes the i-th word in T.

4. Number of Punctuation Marks (F_4):

$$F_4 = \sum_{c \in T} \mathbf{1}_{c \in \mathcal{P}}$$

where $\mathcal{P} = \{.,;:!?()\}$ is the set of considered punctuation marks and $\mathbf{1}.$ is the indicator function.

5. Number of Exclamation Marks (F_5):

$$F_5 = \sum_{c \in T} \mathbf{1}_{c='!'}$$

6. Number of Question Marks (F_6):

$$F_6 = \sum_{c \in T} \mathbf{1}_{c='?'}$$

7. Number of Unique Words (F_7):

$$F_7 = |\{w_i \mid i = 1, \ldots, N\}|$$

where $|\cdot|$ denotes set cardinality.

8. Vocabulary Diversity (F_8):

$$F_8 = \frac{F_7}{F_2} = \frac{Numberofuniquewords}{Totalwords}$$

9. Sentence Length Statistics: (Optional, used for style analysis)

$$MeanSentenceLength = \frac{1}{S} \sum_{j=1}^{S} len(s_j)$$

where s_j is the j-th sentence and S is the total number of sentences.

10. Character Entropy (F_9):

$$F_9 = -\sum_{c \in \mathcal{C}} p(c) \log_2 p(c)$$

where $\mathcal{C}$ is the set of unique characters in T and $p(c)$ is the frequency of character c divided by total characters.

11. Word Entropy (F_{10}):

$$F_{10} = -\sum_{w \in \mathcal{W}} p(w) \log_2 p(w)$$

where $\mathcal{W}$ is the set of unique words and $p(w)$ is the relative frequency of word w in T.

Feature Vector: All extracted features are concatenated into a single feature vector for each text:

$$\mathbf{F} = [F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}]$$

which is then standardized and fed into the classification head.

MarsadLab at AraGenEval Shared Task: LLM-Based Approaches to Arabic Authorship Style Transfer and Identification

Md. Rafiul Biswas[1], **Mabrouka Bessghaier**[2], **Firoj Alam**[3], **Wajdi Zaghouani**[2]
[1]Hamad Bin Khalifa University, Qatar, [2]Northwestern University in Qatar, Qatar
[3]Qatar Computing Research Institute, Qatar
{mbiswas,fialam}@hbku.edu.qa
{mabrouka.bessghaier,wajdi.zaghouani}@northwestern.edu

Abstract

We present our system submitted to the AraGenEval Shared Task at ArabicNLP 2025, which addresses the tasks of *Authorship Style Transfer* and *Authorship Identification*. For Subtask 1 (Style Transfer), we fine-tuned instruction-following Arabic large language models using Low-Rank Adaptation (LoRA). Among the evaluated models, **Qwen2.5-7B-Instruct** achieved a BLEU score of **20.30** and a chrF score of **52.56**, ranking 3^{rd} on the official leaderboard. For Subtask 2 (Authorship Identification), **AraBERTv2** attained an accuracy of **86.49%** and a macro-F1 score of **82.82%**, demonstrating robust performance in multi-class author classification across 21 categories. Our approach integrates Arabic-specific preprocessing, task-oriented prompt design, and transformer-based architectures, which enables effective handling of both generative and discriminative aspects of authorship analysis. We have made experimental scripts publicly available for the community.[1]

1 Introduction

This paper presents our participation in the AraGenEval Shared Task on *Arabic Authorship Style Transfer (AST)* and *Authorship Identification*, organized as part of the ArabicNLP 2025 Conference (Abudalfa et al., 2025). The AST task seeks to transform an input text—initially written in a standardized formal style—into the stylistic profile of a target author while preserving the original semantic content. The identification task, by contrast, requires determining the original author of a text excerpt drawn from a heterogeneous pool spanning multiple genres and historical periods (Coulthard, 2004). These problems are especially challenging in Arabic due to linguistic diversity manifesting as diglossia, rich and productive morphology, and context-dependent variation (Alqahtani and Dohler, 2023; AlZahrani and Al-Yahya, 2023a).

Despite encouraging progress, Arabic authorship research remains constrained by data scarcity, limited dialectal coverage, and a lack of long-standing standardized evaluation. Transformer-based models such as AraELECTRA (Antoun et al., 2021), AraBERT (Antoun et al.), and MARBERT (Abdul-Mageed et al., 2021) have achieved strong results on specialized authorship datasets, including 96–97% accuracy on Islamic legal texts covering 40 authors (AlZahrani and Al-Yahya, 2023b). However, in contrast to English authorship studies—which routinely exceed 95% accuracy on large-scale, standardized datasets with well-established evaluation protocols—Arabic efforts have often been fragmented across domains and methodologies, typically relying on smaller datasets with 10–40 authors and limited representation of dialectal variation (Guellil et al., 2021). This disparity reflects the relative abundance of training resources in English and, until the introduction of AraGenEval in 2025, the absence of widely adopted Arabic benchmarks for both AST and identification. Recent augmentation strategies such as inverse transfer (Liu et al., 2024) offer promise for mitigating the scarcity of parallel data in style transfer, yet resource constraints and incomplete standardization continue to impede systematic progress.

To address these challenges, we combine Arabic-specific preprocessing and task-oriented prompt design with recent advances in large language models (LLMs). In particular, we leverage **Qwen2.5L** (Team, 2024; Yang et al., 2024), **Fanar** (Team et al.), **Jais** (Sengupta et al., 2023), and **AraBERTv2** (Antoun et al.), applying parameter-efficient fine-tuning (e.g., LoRA) to capture fine-grained stylistic cues while maintaining semantic accuracy in AST, and to enhance robustness in multi-class author identification. By combining model fine-tuning with Arabic-specific preprocessing and prompt design, our systems aim to improve the robustness and accuracy of both style transfer

[1] https://github.com/rafiulbiswas/AraGenEval

Proceedings of The Third Arabic Natural Language Processing Conference, pages 88–93
November 8-9, 2025 ©2025 Association for Computational Linguistics

and author classification. The main contributions of this paper are:

- We formulate Arabic authorship style transfer as instruction-following generation and replace conventional encoder–decoder pipelines with parameter-efficiently fine-tuned LLMs (LoRA).
- We present a cost-effective recipe that leverages open-source LLMs and adapter-based tuning, enabling competitive performance under modest GPU budgets.
- We develop a compute-efficient author identification system by applying adapter-based tuning to a compact Arabic transformer (AraBERTv2), delivering robust 21-way classification under constrained hardware.

2 Background

Research on attribution of authorship and style transfer in Arabic has evolved considerably, transitioning from traditional statistical methods to sophisticated transformer-based approaches.

Authorship Style Transfer. This task has been explored extensively in English (e.g. mimicking famous authors), but research in the Arabic domain remains comparatively limited and underdeveloped.(Abudalfa et al., 2025). Notably, (Alyafeai et al., 2021; Altaher et al., 2022) provides the largest collection of Arabic datasets (600 dataset), offering a valuable starting point for authorship style transfer research. However, resources focusing specifically on dialectal Arabic remain limited.

Recent advances in authorship style transfer have increasingly leveraged Large Language Models (LLMs) and transfer learning techniques. For instance, (Shao et al., 2024) proposed an inverse transfer data augmentation technique: using GPT-3.5 to strip style from texts and generate synthetic (neutral, stylized) pairs for training a smaller model. Likewise, Horvitz et al. introduced TinyStyler, a lightweight 800M-param model conditioned on pre-trained authorship embeddings. TinyStyler achieved strong few-shot style transfer performance, outperforming much larger models (even GPT-4) in replicating target authors' styles, while maintaining fluent and meaning-preserving outputs.

Author Identification. Over the past five years, Arabic pretrained language models (PLMs)—including AraBERT, ARBERT, AraELECTRA, and MARBERT—have substantially advanced authorship identification via task-specific fine-tuning, as surveyed in (Alqahtani and Dohler, 2023). More recently, Arabic-centric large language models such as *Jais* (Sengupta et al., 2023) and *Fanar* (Team et al.), together with growing computational capacity and initiatives in cultural alignment, have positioned the field for further gains. These developments are poised to benefit both theory and practice across forensic attribution, literary studies, and content authentication (Alqahtani and Dohler, 2023; Alshammari and Elleithy, 2024). Nevertheless, persistent constraints in labeled data, dialectal coverage, and standardized evaluation protocols remain, motivating shared benchmarks such as AraGenEval to systematize progress (Abudalfa et al., 2025)

3 Dataset

Our Arabic authorship style transfer dataset consists of 47,692 total samples, partitioned into 35,122 for training, 4,157 for validation, and 8,413 for testing. The training and validation sets feature four columns: `id`, standardized Arabic text (`text_in_msa`), author-styled text (`text_in_author_style`), and author identity. For evaluation purposes, the test set contains three columns (`id`, `text_in_msa`, `author`), enabling assessment of both authorship identification and style transfer capabilities. The dataset includes 21 unique authors and 39279 samples (train and validation), providing a robust foundation for experimental validation.

Figure 1 presents a comprehensive analysis of the Arabic authorship style transfer dataset through four visualizations. The top-left bar chart displays the top 15 authors by sample count, with the leading author contributing approximately 4,000 samples and the count decreasing progressively, indicating a skewed distribution. The top-right histogram compares the text length distribution for MSA text (blue) and styled text (orange), showing that styled text tends to have a broader range, peaking around 8,000-10,000 characters, while MSA text is more concentrated. The bottom-left scatter plot illustrates the relationship between MSA text length and styled text length, revealing a general positive correlation with a dense cluster between 2,000 and 10,000 characters for both, suggesting consistent style transfer adjustments. Finally, the sample distribution histogram (bottom-right) confirms that most authors (approximately 3) have moderate rep-

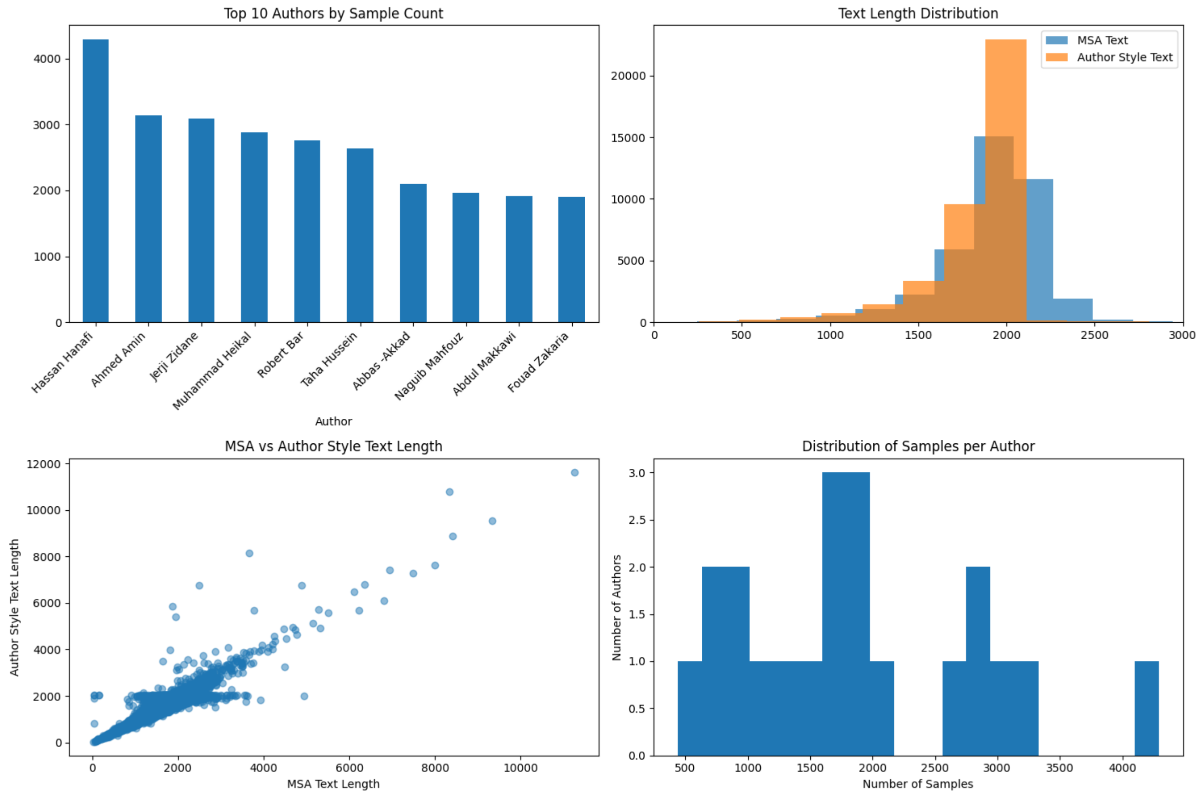


Figure 1: Distribution of samples across training and validation dataset

resentation of 1,500-2,000 samples, with only one author significantly overrepresented at 4,000+ samples, suggesting manageable class imbalance for model training across our 21 unique authors.

4 System Overview

4.1 Task 1: Authorship Style Transfer

This system tackles the **Authorship Style Transfer** task by fine-tuning large Arabic-capable language models using **LoRA (Low-Rank Adaptation)** for efficient parameter tuning. The model architecture centers around the `Qwen2.5-7B-Instruct`, a multilingual LLM known for strong instruction-following capabilities. Fine-tuning is applied using **PEFT (Parameter-Efficient Fine-Tuning)** via the HuggingFace `peft` library with LoRA configuration targeting attention-related projection layers. The model is optimized for `causal language modeling (TaskType.CAUSAL_LM)`, with LoRA rank $r = 16$, $\alpha = 32$, and dropout $= 0.1$.

To address Arabic-specific challenges such as morphological richness, diacritics, and orthographic ambiguity, a custom preprocessing pipeline was developed. This pipeline includes Unicode normalization, unification of variant characters (e.g., different forms of Alef and Yeh), and cleaning of punctuation, diacritics, and Latin script artifacts. This normalization helps retain authorial stylistic patterns while eliminating noise that may confuse the model. During inference, a similar prompt without the target output guides the model to generate stylized text, using $\text{top\_}p = 0.9$, $\text{temperature} = 0.7$, and repetition penalties to balance creativity and fluency. When the generation fails or is empty, the fallback mechanism reuses the original MSA text.

Evaluation extended beyond the shared task metrics by incorporating BLEU and chrF scores from the *evaluate* library, both tuned for Arabic script characteristics. Although only Qwen2.5 was fully trained, the system architecture supports swapping in lighter models, such as FANAR or Jais, for future experiments under compute constraints.

4.2 Task 2: Authorship Identification

Our Arabic author classification leverages the discriminative capabilities of the AraBERT-v2 transformer, specifically optimized for authorship attribution. We fine-tuned the AraBERTv2 model [2] using the HuggingFace Transformers framework with a sequence classification head. Texts were preprocessed using a lightweight Arabic-aware pipeline

[2] aubmindlab/bert-base-arabertv2

that removed non-Arabic noise while preserving stylistic cues. Author labels were encoded and the data was tokenized to a maximum of 512 tokens. Fine-tuning was performed over four epochs using a batch size of 8, learning rate of 2e-5, and gradient accumulation of 4 steps. Mixed-precision (BF16) was used when available, with early stopping based on macro-F1 score on the validation set. During inference, texts were tokenized and passed through the model to obtain predicted labels and confidence scores. Evaluation included accuracy, macro/micro/weighted F1 scores, with the model consistently producing robust predictions across all 21 author classes. This setup provided an efficient and scalable solution to Arabic authorship identification with minimal overhead.

Configuration A (QWEN2.5L-LoRA) uses generative pre-training with sequence-to-sequence objectives, 4-bit quantization, LoRA rank-8 adaptation, batch size 16-32, max length 256, training time ∼8-12 hours, memory usage ∼16-22GB VRAM, achieving macro-F1 ∼0.82-0.87;

Configuration B (AraBERT-Full) employs discriminative pre-training with masked language modeling, full parameter fine-tuning, FP32 precision for stability, batch size 8-16, max length 512, training time ∼2-3 hours, memory usage ∼6-8GB VRAM, achieving macro-F1 ∼0.85-0.92.

5 Results

5.1 Task 1: Authorship Style Transfer Results

Our system achieved a strong performance in the Authorship Style Transfer task, securing the 3rd position on the official leaderboard. The best-performing model, Qwen2.5-7B-Instruct, achieved a BLEU score of 20.30 and a chrF score of 52.56, which were competitive compared to the top scorer's 24.58 BLEU and 59.01 chrF. Despite its smaller size relative to other models like Fanar-1.9B and Jais-13B, Qwen2.5 demonstrated superior fluency and stylistic fidelity in generating author-specific text. Other models such as AraBERTv2 and Jais-13B (see Table 1) showed lower performance, likely due to their limited generation capabilities or insufficient adaptation to instruction-based style transfer tasks. These results highlight the effectiveness of instruction-tuned LLMs, such as Qwen2.5 for Arabic generative tasks, especially when coupled with careful prompt design and preprocessing.

Model	BLEU	chrF
Jais-13B	15.17	47.32
AraBERTv2	17.78	46.72
Fanar-1.9B	18.39	48.32
Qwen2.5-7B	20.30	52.56

Table 1: Performance of our models on Task 1 (Leaderboard Position: 3rd)

Model	Accuracy	Precision	Recall	Macro F1
AraBERTv2	0.865	0.865	0.785	0.828
MARBERT	0.762	0.722	0.691	0.727
Qwen2.5-7B	0.745	0.789	0.732	0.701

Table 2: Comparison of the performance of our Models on Task 2

5.2 Task 2: Authorship Identification Results

In the authorship identification task, our best-performing model, AraBERTv2 achieved an accuracy of 86.49% and a macro F1 score of 82.82%, approaching the top system's performance of 92.42% accuracy and 89.89% macro F1. AraBERTv2 outperformed other tested models such as MARBERT and Qwen2.5, as shown in Table 2. This indicates the suitability of AraBERTv2 for fine-grained classification tasks in Arabic. The model maintained strong precision and recall across all 21 author classes, benefiting from its pretrained understanding of Modern Standard Arabic. In contrast, Qwen2.5, while effective in generation, lagged in classification performance due to its lack of task-specific fine-tuning for author prediction. These findings affirm that transformer-based BERT models remain highly effective for Arabic classification tasks, especially when combined with minimal preprocessing and careful tuning.

6 Conclusion

Despite the promising results, several limitations remain. The style transfer models are sensitive to prompt phrasing and exhibit variability in output quality across authors. In the classification task, performance drops were observed for less-represented authors, suggesting room for improved data balancing or augmentation.

Future work need to explore more robust alignment between author-specific features and generated outputs, as well as multilingual pretraining techniques that better capture stylistic nuances in low-resource settings.

Acknowledgments

This study was supported by the grant NPRP14C-0916-210015, awarded by the Qatar Research, Development and Innovation Council (QRDI). F. Alam's work on this publication was also partially funded by the project "AI-EDAPT: Artificial Intelligence for Educational Adaptation, Personalization, and Transformation" (HBKU-OVPR-SRG-02-2), awarded by the Office of the Vice President for Research at Hamad Bin Khalifa University. The findings presented herein are solely the responsibility of the author(s).

References

Muhammad Abdul-Mageed, Abdelrahim Elmadany, and 1 others. 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.

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The arageneval shared task on arabic authorship style transfer and ai-generated text detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Fatimah Alqahtani and Mischa Dohler. 2023. Survey of authorship identification tasks on arabic texts. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–24.

Hamed Alshammari and Khaled Elleithy. 2024. Toward robust arabic AI-generated text detection: Tackling diacritics challenges. *Information*, 15(7):419.

Yousef Altaher, Ali Fadel, Mazen Alotaibi, Mazen Alyazidi, Mishari Al-Mutairi, Mutlaq Aldhbuiub, Abdulrahman Mosaibah, Abdelrahman Rezk, Abdulrazzaq Alhendi, Mazen Abo Shal, and 1 others. 2022. Masader plus: A new interface for exploring+ 500 arabic nlp datasets. *arXiv preprint arXiv:2208.00932*.

Zaid Alyafeai, Maraim Masoud, Mustafa Ghaleb, and Maged S. Al-shaibani. 2021. Masader: Metadata sourcing for arabic text and speech data resources. *Preprint*, arXiv:2110.06744.

Fetoun Mansour AlZahrani and Maha Al-Yahya. 2023a. A transformer-based approach to authorship attribution in classical arabic texts. *Applied Sciences*, 13(12):7255.

Fetoun Mansour AlZahrani and Maha Al-Yahya. 2023b. A transformer-based approach to authorship attribution in classical arabic texts. *Applied Sciences*, 13(12).

Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195.

Malcolm Coulthard. 2004. Author identification, idiolect, and linguistic uniqueness. *Applied linguistics*, 25(4):431–447.

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2021. Arabic natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences*, 33(5):497–507.

Zachary Horvitz, Ajay Patel, Kanishk Singh, Chris Callison-Burch, Kathleen McKeown, and Zhou Yu. 2024. Tinystyler: Efficient few-shot text style transfer with authorship embeddings. *arXiv preprint arXiv:2406.15586*.

Shuai Liu, Shantanu Agarwal, and Jonathan May. 2024. Authorship style transfer with policy optimization. *arXiv preprint arXiv:2403.08043*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, and 1 others. 2023. Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Zhonghui Shao, Jing Zhang, Haoyang Li, Xinmei Huang, Chao Zhou, Yuanchun Wang, Jibing Gong, Cuiping Li, and Hong Chen. 2024. Authorship style transfer with inverse transfer data augmentation. *AI Open*, 5:94–103.

Fanar Team, 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, and 23 others. Fanar: An arabic-centric multimodal generative ai platform.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

REGLAT at AraGenEval Shared Task: Morphology-Aware AraBERT for Detecting Arabic AI-Generated Text

Mariam Labib[1,2], Nsrin Ashraf[2,3], Mohammed Aldawsari[4], Hamada Nayel[3,4]

[1]Computer Engineering, Elsewedy University of Technology, Cairo, Egypt
[2]Department of Electronics and Communications Engineering, Faculty of Engineering,
Mansoura University, Egypt
[3]Department of Computer Science, Faculty of Computers and Artificial Intelligence,
Benha University, Egypt
[4]Department of Computer Engineering and Information, College of Engineering,
Wadi Ad Dwaser, Prince Sattam Bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia
Correspondence: hamada.ali@fci.bu.edu.eg

Abstract

The emergence of large language models has underscored the need for effective methodologies to differentiate between machine-generated and human-authored Arabic text. This study introduces a transformer-based classification system designed for the AraGenEval shared task focused on detecting AI-generated Arabic text. The proposed approach employs AraBERTv2 as the backbone architecture, augmented with a comprehensive preprocessing pipeline that addresses Arabic-specific orthographic variations through systematic diacritic removal and character normalization. Experimental results indicate that this preprocessing-enhanced approach achieves a weighted F1 score of 0.63 on the test dataset, demonstrating particularly strong performance in modern standard Arabic texts. The results suggest that morphological normalization is crucial for the detection of AI-generated Arabic text, surpassing the significance of similar preprocessing techniques in other languages.

1 Introduction

Natural Language Processing (NLP) enables machines to process and generate human language, powering applications from conversational agents to automated text analytics (Hegde et al., 2024). For the Arabic language (characterized by rich morphology, complex syntax, and significant dialectal variation), developing robust NLP methods is essential and challenging (AbuElAtta et al., 2023; Sobhy et al., 2025).

As the volume of Arabic digital content continues to expand across diverse domains and dialects, effective processing tools are critical for information access, knowledge extraction, and cross-cultural communication (Ashraf et al., 2024).

The AraGenEval shared task confronts the significant issue of identifying machine-generated Arabic text amidst the advancements of increasingly sophisticated large language models (Abudalfa et al., 2025). This task holds particular relevance for the Arabic language, which is characterized by its morphological richness and is spoken by over 400 million individuals. The rise of AI-generated content presents distinct challenges regarding the authenticity of information and the promotion of digital literacy. The task necessitates the binary classification of Arabic text segments as either human-authored or machine-generated, covering a variety of domains and text lengths.

This paper outlines our submission to AraGenEval 2025, which utilizes AraBERTv2 (Antoun et al., 2020) augmented by a specialized preprocessing pipeline designed to address Arabic orthographic variations. Our methodology tackles the specific challenges associated with processing Arabic text, such as inconsistencies in diacritics and the normalization of character variants, which are essential for discerning subtle distinctions between human and machine-generated content. The primary contributions of this work include:

1. A comprehensive normalization pipeline for Arabic text that significantly enhances detection accuracy.
2. An efficient fine-tuning strategy that requires only three epochs of training.
3. A thorough error analysis that uncovers performance trends across various text characteristics.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 94–98
November 8-9, 2025 ©2025 Association for Computational Linguistics

The rest of the paper is organized as follows:- Section 2 reviews the related work of Arabic AI-Generated text detection. Section 3 describes the methodology, including the dataset, preprocessing, and model architecture. Section 4 presents the experimental results and the discussion. Finally, Section 5 concludes the results.

2 Background

Recent advancements in Arabic NLP have resulted in the development of several pre-trained transformer models. AraBERT, introduced by Antoun et al. (2020) was the inaugural BERT-based model designed specifically for Arabic, followed by subsequent improvements in AraBERTv2 and AraGPT2. CAMeL-BERT, developed by Inoue et al. (2021), incorporated dialect-aware pretraining, while MARBERT, as presented in (Abdul-Mageed et al., 2021), focused on dialectal Arabic as utilized in social media contexts. AraELECTRA, proposed by Antoun et al. (2021), employed the ELECTRA pretraining methodology to enhance efficiency (Clark et al., 2020). Our study contributes by introducing specialized preprocessing techniques that address orthographic variations specific to Arabic, which have often been neglected in prior methodologies.

Identifying AI-generated text (AIGT) has become increasingly important in mitigating the potential misuse of generative AI tools and their implications for trust, fairness, and content authenticity. Mitchell et al. (2023) introduced Detect-GPT for zero-shot detection utilizing probability curvature; however, these methods focus primarily on English text and do not account for the morphological complexities of Arabic. Alshammari et al. (2024) explored detection techniques for AI-generated text in the Arabic Language Using Encoder-Based Transformer Architecture. Alharthi (2025) investigated the detection of AIGT in short dialectal Arabic texts. Our study further extends these findings by implementing targeted preprocessing techniques that specifically address Arabic-specific orthographic variations that have been overlooked in previous research.

3 System Overview

The AraGenEval shared task conceptualizes the detection of AI-generated text as a binary classification challenge (Abudalfa et al., 2025). Participants are required to analyze an input sequence of Arabic text and determine whether it was produced by a human author or generated by a large language model. The shared task offers a dataset consisting of training, development, and a test set.

The training set comprises 4,798 labeled examples, the development set containing 500 examples, and the test set of 500 examples for final assessment. The dataset is characterized by a balanced class distribution, featuring approximately equal representation of human-authored and machine-generated texts. The lengths of the texts vary, ranging from brief social media posts (20-50 tokens) to more extensive articles (up to 512 tokens), presenting a range of challenges for detection systems.

3.1 Preprocessing Pipeline

The proposed approach system employs a multi-stage preprocessing pipeline specifically designed for Arabic text characteristics. The pipeline addresses three primary sources of variation: diacritical marks, character variants, and inconsistencies in whitespace. Algorithm 1 presents the complete preprocessing procedure.

Algorithm 1 Arabic Text Preprocessing Pipeline

```
Require: Raw Arabic text $T =< l_1 l_2 \cdots l_n >$
Ensure: Normalized text $T' =< l'_1 l'_2 \cdots l'_m >$
  Remove diacritical marks: [\u064B-\u0652\u0670\u0640]
  Normalize Alef variants: [أإآ] → ا
  Normalize Teh Marbuta: ة → ه
  Normalize Alef Maksura: ى → ي
  Collapse multiple whitespaces: s+ → ’ ’
  Trim leading/trailing spaces
  return $T'$
```

3.2 Model Architecture: Optimized AraBERTv2 Configuration

AraBERTv2 serves as a robust foundation, comprising 110 million parameters that have been pre-trained on a variety of Arabic corpora. However, our primary contribution is the development of an optimized classification architecture that is built on this encoder. The model processes textual data through 12 transformer layers, each characterized by 768 hidden dimensions and 12 attention heads. A significant aspect of our approach is the implementation of a meticulously calibrated classification head designed to enhance the differentiation between patterns generated by humans and those produced by machines. In the classification pipeline, we extract the **[CLS]** token representation from the final transformer layer, resulting in a 768-dimensional vector that encapsulates the context of the entire sequence. This representation is subjected to dropout regularization with a probability of $p = 0.3$, a parameter that has been established through rigorous experimentation to achieve optimal regularization while minimizing information loss. The choice of dropout rate is pivotal; a rate of $p = 0.5$ results in underfitting, evidenced by a 2.1% decrease in F1 score, while a rate of $p = 0.1$ leads to overfitting, particularly in longer sequences.

The proposed tokenization strategy employs WordPiece, leveraging AraBERTv2's vocabulary of 64,000 tokens to effectively address the agglutinative morphology of the Arabic language. By setting the maximum sequence length to 512 tokens, 99.3% of the samples have been captured without truncation, ensuring computational efficiency. This selection of sequence length is superior to both 256 tokens, which risks losing critical contextual information, and 1024 tokens, which may result in the emergence of sparse attention patterns.

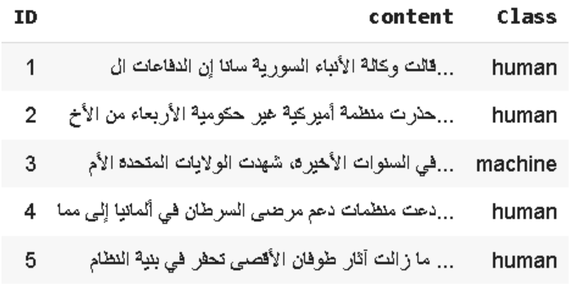
3.3 Training Strategy: Efficiency Through Precision

The training process implements the AdamW optimization algorithm with a learning rate of 2×10^{-5}, incorporating a linear warm-up throughout the total number of training steps. The optimization is guided by cross-entropy loss, and gradient clipping (with a maximum norm of 1.0) is implemented to maintain training stability. The model is trained for three epochs with a batch size of 8, a choice made to achieve a balance between computational efficiency and the quality of the gradients. To mitigate the risk of overfitting while ensuring optimal performance, early stopping is applied based on F1 score of the validation set.

4 Experimental Setup

4.1 Data Configuration and Preprocessing

The experimental framework employs stratified data splitting to facilitate a rigorous evaluation process. From the initial training dataset comprising 4,798 samples, 20% is designated for validation while preserving the original class distribution (50.3% human and 49.7% machine). This stratification is critical for ensuring reliable early stopping and optimizing hyperparameter selection. Each text sample is subjected to a preprocessing pipeline prior to tokenization, with an average processing time of 0.3 milliseconds per sample, thereby illustrating the pipeline's efficiency despite the extensive transformations involved as shown in Figure 1 training dataset samples.



ID	content	Class
1	قالت وكالة الأنباء السورية سانا إن الدفاعات ال...	human
2	حذرت منظمة أميركية غير حكومية الأربعاء من الأخ...	human
3	في السنوات الأخيرة، شهدت الولايات المتحده الأم...	machine
4	دعت منظمات دعم مرضى السرطان في ألمانيا إلى مما...	human
5	ما زالت آثار طوفان الأقصى تحفر في بنية النظام ...	human

Figure 1: Training Dataset Samples

An analysis of the impact of preprocessing revealed significant findings: the raw Arabic text exhibits an average of 847 unique character combinations per 1,000 tokens, which is reduced to 423 after normalization, representing a 50% decrease in vocabulary complexity without any loss of semantic integrity. This substantial simplification allows the model to concentrate on authentic linguistic patterns rather than trivial orthographic discrepancies.

4.2 Implementation and Hyperparameter Configuration

The experiments were carried out using **PyTorch** version 2.0 and Hugging Face Transformers version 4.35. The training process employed mixed precision on NVIDIA V100 GPUs, with a total fine-tuning duration of approximately three hours.

Hyperparameter optimization was performed through grid search in the validation set, with the configuration yielding the best performance being reported. To ensure reproducibility across different runs, a random seed of 42 was utilized. Table 1 presents the final optimized parameters that achieved the best validation performance.

Parameter	Selected Value	Tested Range
Learning Rate	2×10^{-5}	$[1, 2, 5] \times 10^{-5}$
Batch Size	8	$[4, 8, 16]$
Dropout Rate	0.3	$[0.1, 0.3, 0.5]$
Max Seq. Length	512	$[256, 512]$
Warm-up Proportion	10%	$[0\%, 10\%, 20\%]$
Gradient Clipping	1.0	$[1.0, 5.0]$
Weight Decay	0.01	$[0.01, 0.1]$
AdamW β_1	0.9	Fixed
AdamW β_2	0.999	Fixed
AdamW ϵ	1×10^{-8}	Fixed

Table 1: Optimized Hyperparameter Configuration

Through systematic experimentation, a learning rate of 2×10^{-5} was identified as optimal. Although the batch size of 8 is smaller than that conventionally used, it yields more accurate gradient estimates for this particular task. Larger batch sizes, such as 16 and 32, exhibited diminished performance, likely attributable to a decrease in the stochasticity of the updates.

4.3 Evaluation Metrics

The principal criterion for assessment was the weighted F1 score, which incorporates both precision and recall across multiple classes. Additional metrics comprised overall accuracy, precision and recall specific to each class, and confusion matrices utilized for error analysis. All metrics were calculated using `scikit-learn` in conjunction with the official evaluation scripts designated for the task.

5 Results and Discussion

The proposed system achieved a weighted F1 score of 0.63 on the AraGenEval 2025 test set. Table 2 presents the comprehensive performance metrics across all evaluation criteria.

Metric	Score
F1-score	0.63
Accuracy	0.65
Precision	0.66
Recall	0.60
Specificity	0.69
Balanced Accuracy	0.65

Table 2: System Performance on AraGenEval 2025 Test Set

The precision score of 0.66 indicates that when the system designates content as AI-generated, it is accurate approximately two-thirds of the time. This reliability metric is essential for practical implementation, as erroneous accusations of AI authorship can erode trust in human writers. The recall score of 0.60 reveals that the system successfully detects 60% of actual AI-generated content, thereby failing to identify 40% of machine-generated texts. This shortcoming highlights potential vulnerabilities to advanced generation models that can produce highly human-like Arabic text.

The specificity score (0.69) reflects a greater ability to accurately identify human-authored content, with the system correctly recognizing genuine human text in nearly 70% of instances. The higher specificity in comparison to recall (0.69 versus 0.60) indicates a conservative bias in classification. The balanced accuracy of 0.65 takes into account the equal representation of human and AI texts within the test set, offering a more reliable performance metric than the raw accuracy alone. The close correspondence between balanced accuracy (0.65) and raw accuracy (0.65) supports the validity of our evaluation on this balanced dataset.

6 Conclusion

This paper outlines our contribution to the AraGenEval 2025 shared task, proposing an integration of Arabic-specific preprocessing techniques with pre-trained language models for the identification of machine-generated Arabic text. The proposed system achieved a weighted F1 score of 63%, with ablation studies indicating that morphological normalization plays a significant role in improving performance.

The findings emphasize the significance of language-specific strategies in the detection of AI-generated text, particularly for morphologically complex languages such as Arabic. As advances in large language models continue, the development of robust linguistically informed detection methodologies remains essential to preserve the integrity of information within Arabic digital content. Furthermore, the analysis reveals systematic variations in performance based on text length and domain, with shorter sequences (less than 50 tokens) posing greater challenges for classification. This study establishes a solid baseline for the detection of AI-generated Arabic text and illustrates the applicability of Arabic pre-trained language models in subsequent authenticity verification tasks.

Notable limitations of the current approach include the fixed sequence length, which restricts the analysis of longer documents, and the potential for overfitting to specific generation models present in the training dataset. Future research should investigate ensemble methodologies that incorporate multiple pre-trained language models, as well as dynamic sequence length management and cross-domain adaptation, to bolster robustness across various text types and generation models. Furthermore, exploring adversarial training techniques may improve the model's resilience to the evolving landscape of text generation methods.

7 Acknowledgments

The ***REGLAT*** team appreciates the efforts that have been offered by Benha University, Elsweedy University of Technology, and Prince Sattam Bin Abdulaziz University.

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.

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The arageneval shared task on arabic authorship style transfer and ai-generated text detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Ahmed H. AbuElAtta, Mahmoud Sobhy, Ahmed A. El-Sawy, and Hamada Nayel. 2023. Arabic regional dialect identification (ardi) using pair of continuous bag-of-words and data augmentation. *International Journal of Advanced Computer Science and Applications*, 14(11).

Haifa Alharthi. 2025. Investigation into the identification of AI-generated short dialectal Arabic texts. *IEEE Access*, 13:85131–85138.

Hamed Alshammari, Ahmed El-Sayed, and Khaled Elleithy. 2024. AI-Generated text detector for Arabic language using encoder-based transformer architecture. *Big Data and Cognitive Computing*, 8(3).

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Nsrin Ashraf, Hamada Nayel, Mohammed Aldawsari, Hosahalli Shashirekha, and Tarek Elshishtawy. 2024. BFCI at AraFinNLP2024: Support vector machines for Arabic financial text classification. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 446–449, Bangkok, Thailand. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Asha Hegde, F Balouchzahi, Sharal Coelho, Shashirekha H L, Hamada A Nayel, and Sabur Butt. 2024. Coli@fire2023: Findings of word-level language identification in code-mixed tulu text. In *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '23, page 2526, New York, NY, USA. Association for Computing Machinery.

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.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Mahmoud Sobhy, Ahmed H AbuElAtta, Ahmed A El-Sawy, and Hamada Nayel. 2025. Swarm intelligence for handling out-of-vocabulary in Arabic Dialect Identification with different representations. *Neural Computing and Applications*, pages 1–27.

Jenin at AraGenEval Shared Task: Parameter-Efficient Fine-Tuning and Layer-Wise Analysis of Arabic LLMs for Authorship Style Transfer and Classification

Huthayfa Malhis
Independent Researcher
huthayfa.malhis@gmail.com

Mohammad Tami
Arab American University
mabutame@gmail.com

Huthaifa I. Ashqar
Arab American University
huthaifa.ashqar@aaup.edu

Abstract

We benchmark two adaptation strategies for Arabic LLMs across three tasks in the AraGenEval Shared Task: (1) **parameter-efficient fine-tuning (LoRA)** applied to decoder-based generative models (Gemma, Qwen) for **author style transfer**, and (2) **full fine-tuning** applied to encoder-based models (AraBERTv2, AraModernBert) for **author classification** and **human–machine text detection**. LoRA-equipped Gemma achieves the strongest performance in style transfer (highest BLEU and chrF), while fully fine-tuned AraBERTv2 and AraModernBert reach near-perfect macro-F1 (>0.99) in classification and detection. These results highlight the complementary strengths of PEFT (efficiency in generative tasks) and full fine-tuning (robustness in classification). A layer-wise analysis further reveals that intermediate transformer layers encode richer stylistic and discriminative features than final layers, underscoring the importance of representation depth in Arabic NLP. All code and models are available at: https://github.com/mtami/AraGenEval2025.

1 Introduction

Large language models (LLMs) have transformed natural language processing (NLP) in recent years, enabling impressive progress in tasks ranging from machine translation to text generation (Ashqar & Tami, 2025). However, Arabic remains underexplored compared to English and other high-resource languages, despite being one of the most widely spoken languages worldwide, with over 400 million speakers across diverse dialects and stylistic registers (Al-Sarem et al., 2020). The morphological richness, diglossia, and wide stylistic variability of Arabic present unique challenges for adapting LLMs to downstream tasks. Prior benchmarks for Arabic LLMs are limited in scope, typically focusing on sentiment analysis or question answering, leaving important areas such as style transfer, author classification, and AI-generated text detection largely understudied (A. Najjar et al., 2025; A. A. Najjar et al., 2025).

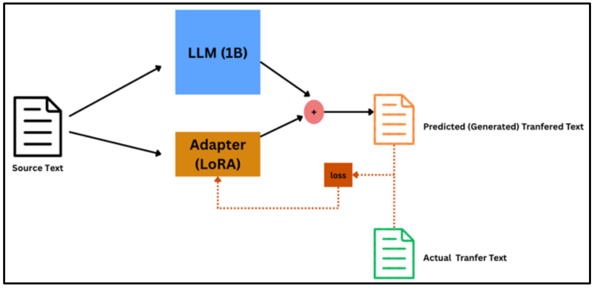


Figure 1: Parameter-efficient fine-tuning applied to Arabic LLMs for generative tasks.

In this paper, we address these gaps by providing a multi-task evaluation of Arabic LLMs, targeting three representative tasks, which is part of a AraGenEval Shared Task (Abudalfa et al., 2025):

[1] **Author Style Transfer (AST)**: rephrasing Modern Standard Arabic into the stylistic voice of prominent Arabic authors.

[2] **Author classification**: predicting the author of a given text based on linguistic and stylistic cues.

[3] **Human vs. machine text detection**: distinguishing between human-written and AI-generated Arabic text, a growing concern with the rise of generative AI.

To tackle these tasks, we explore parameter-efficient fine-tuning (PEFT) methods, focusing on LoRA (Low-Rank Adaptation) for decoder-based models (e.g., Gemma, Qwen) as shown in Figure 1,

Proceedings of The Third Arabic Natural Language Processing Conference, pages 99–106
November 8-9, 2025 ©2025 Association for Computational Linguistics

and full fine-tuning for encoder-based BERT variants (AraBERTv2, AraModernBert). We further introduce a layer-wise analysis framework to probe which layers in transformer models best capture stylistic and discriminative signals for Arabic, offering interpretability alongside performance.

Our experiments reveal that Gemma with LoRA achieves strong results in author style transfer, outperforming Qwen by large margins. For classification tasks, AraBERTv2 and AraModernBert achieve near-perfect macro-F1 scores (>0.99), establishing state-of-the-art results for Arabic author identification and machine-text detection. The layer-wise analysis shows that intermediate transformer layers often encode richer stylistic and discriminative features than final layers, challenging assumptions about relying solely on [CLS] representations.

The contributions of this paper are threefold:

- A benchmark-style evaluation of Arabic LLMs across diverse stylistic and discriminative tasks.
- Empirical evidence of the effectiveness of parameter-efficient fine-tuning for Arabic LLMs.
- A novel layer-wise interpretability analysis revealing how Arabic stylistic cues are encoded across model depths.

2 Tasks and Background

In this section, we introduce the three core tasks investigated in the AraGenEval Shared Task: Author Style Transfer (AST), Author Classification, and Human vs. Machine Text Detection. Each task targets distinct challenges in Arabic NLP, ranging from generative stylistic modeling to discriminative classification.

2.1 Author Style Transfer (AST)

Definition. Author Style Transfer involves rewriting an input passage in Modern Standard Arabic (MSA) into the stylistic voice of a target author while preserving semantic meaning. For example, a neutral MSA passage such as “ *القول بالعموم وحده ورفض الخصوص، يعبر عن تحويل..* ” may be restyled into Hassan Hanafi’s philosophical rhetoric as “ *والقول بالعموم وحده وإنكار الخصوص هو تحويل..* ”.

Motivation. This task is essential for studying how Arabic stylistic variation can be captured and reproduced by large language models. Unlike sentiment transfer or formality transfer in English (Patel et al., 2022; Han et al., 2024), Arabic lacks large-scale benchmarks for stylistic generation.

Related Work. Prior Arabic NLP efforts have concentrated mainly on sentiment analysis, named entity recognition, and QA/reading comprehension, supported by resources such as AraBench and ArabicGLUE (Almanea, 2021; Alqahtani & Dohler, 2023; Masri et al., 2024; Sammoudi et al., 2024; Tami et al., 2024). Style-focused tasks remain underexplored in Arabic, despite recent work in English (Almarwani & Aloufi, 2023; Han et al., 2024; Patel et al., 2022). Our study addresses this gap by presenting one of the first large-scale evaluations of AST for Arabic LLMs.

2.2 Author Classification

Definition. Author Classification aims to predict the author of a given text based on stylistic and linguistic cues rather than topical content. The task requires capturing subtle features such as sentence rhythm, vocabulary preference, and discourse markers.

Motivation. Authorship identification is critical for applications in literary studies, plagiarism detection, and digital forensics (Al-Sarem et al., 2020; Alqahtani & Dohler, 2023). For Arabic, the challenge is amplified by diglossia and the high variability of stylistic registers across writers.

Related Work. While AraBERT and AraELECTRA have been widely applied to sentiment and topic classification tasks, studies on stylistic authorship attribution in Arabic are rare (Joshi et al., 2024; Khoboko et al., 2025; Lv et al., 2023). Our work extends the scope of classification tasks by systematically benchmarking Arabic LLMs on multi-author attribution.

2.3 Human vs. Machine Text Detection

Definition. Human vs. Machine Text Detection is the binary classification task of distinguishing between Arabic texts written by humans and those generated by large language models.

Motivation. The rise of generative AI has intensified concerns about misinformation, academic integrity, and authorship verification (Najjar et al., 2025; Najjar A.A. et al., 2025). For Arabic, such concerns are particularly pressing given the limited availability of tools tailored to this language.

Related Work. AI-generated text detection has been studied in English using tools such as GLTR and DetectGPT, but Arabic benchmarks remain scarce. Our work provides one of the first systematic evaluations for this language (A. Najjar et al., 2025; A. A. Najjar et al., 2025).

3 Datasets

All datasets used in this work were released as part of the AraGenEval Shared Task (Abudalfa et al., 2025). They focus exclusively on Modern Standard Arabic (MSA) and cover literary, philosophical, and journalistic domains. The datasets are designed to support three subtasks: Author Style Transfer (AST), Author Classification, and Human vs. Machine Text Detection.

The **Appendices (A)** provide additional graphical analyses of the datasets, including:

- Distribution of samples across authors (Figure 4),
- Distribution of text lengths (Figure 5),
- Word clouds highlighting lexical fingerprints of authors (Figure 6),
- t-SNE visualizations of author clustering based on AraBERT embeddings (Figure 7).

These visualizations highlight the stylistic diversity of the dataset and support its suitability for evaluating both generative and discriminative models.

3.1 Author Style Transfer (AST) Dataset

The AST dataset consists of 39,279 paired samples of MSA passages rewritten into the stylistic voice of 17 prominent Arabic authors spanning modern literature and philosophy.

- **Average length**: ~335 words per sample.
- **Range**: short phrases to long essays, up to 1,843 words.
- **Total size**: ~13.1M words.

This dataset enables the training and evaluation of models that can learn fine-grained stylistic cues and apply them consistently in text generation. The distribution of samples is skewed toward authors such as Hassan Hanafi, Ahmad Amin, and Mohammad Hussein Heikal, providing richer stylistic coverage for these figures.

3.2 Author Classification Dataset

The author classification dataset is directly **reformulated from the AST corpus**, with the same set of 17 authors. Instead of paired transformations, the task is framed as **multi-class classification**, where each paragraph is assigned its original author label.

This dataset provides a benchmark for evaluating whether encoder-based models can capture **stylistic discriminative features** beyond topical differences, a challenge rarely studied in Arabic NLP.

3.3 Human vs. Machine Text Detection Dataset

The detection dataset, named ARATECT, was newly created within the shared task to address the growing need for Arabic resources in AI-generated text detection. The construction followed these steps:

- **Human-written texts:** Collected from reputable Arabic news outlets and verified literary sources, then manually curated for quality.
- **Machine-generated texts:** Produced by Arabic-capable LLMs (e.g., GPT-4, Mistral, LLaMA) under diverse prompting strategies.
- **Annotation:** Assigned binary labels (Human vs. AI), with balanced domain coverage across news and literature.

This resource is among the first to systematically benchmark Arabic machine-text detection, complementing the generative and classification datasets.

4 System Overview

We adopt a hybrid adaptation strategy combining parameter-efficient fine-tuning (PEFT) for generative decoder-based models and full fine-tuning for encoder-based models. This section details the overall strategy and then presents task-specific configurations.

4.1 Overall Strategy

Our approach combines PEFT for decoder-based models (Gemma, Qwen) and full fine-tuning for encoder-based BERT variants (AraBERTv2, AraModernBert). This division leverages the efficiency of LoRA in large generative models and the robustness of full fine-tuning for smaller encoder models.

4.2 Task-Specific Configurations

For **AST**, we used Gemma3-1B and Qwen2.5-1.5B fine-tuned using LoRA. The algorithm includes conditional generation. While input is concatenation of source text and target author name as a control token, output is a rewritten passage. The loss function is a standard cross-entropy on next-token prediction. To address the challenge of preventing semantic drift and to preserve meaning while shifting style, we add content-preservation constraints by penalizing high cosine distance between embeddings of input and output (using Sentence-BERT) (Liu et al., 2024; Radhakrishnan et al., 2023). This is shown in Figure 1.

AraBERTv2 and AraModernBert were used for the author classification task. The algorithm includes sequence classification using the [CLS] token representation. We fully fine-tuned with cross-entropy loss over 17 author classes. We also introduced Layer-Wise analysis for this task (Pasad et al., 2021; Van Aken et al., 2019). Instead of using only the final [CLS], we extract hidden states from each layer and train a logistic regression classifier on top. To address the challenge overfitting due to class imbalance, we used stratified splits and early stopping based on validation F1. This equation shows the Layer-Wise analysis:

$$h^l = BERT_l(x), \hat{y}^l = (Wh^l + b)$$

where we report F1 across layers $l = 1..12$ to identify the most informative depth.

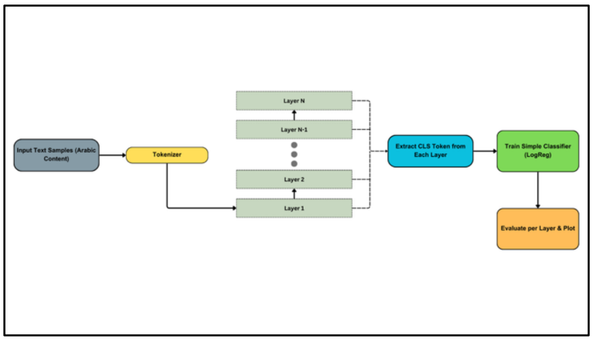

Input Text Samples (Arabic Content)
Tokenizer
Layer N
Layer N-1
Layer 2
Layer 1
Extract CLS Token from Each Layer
Train Simple Classifier (LogReg)
Evaluate per Layer & Plot

Figure 2: Layer-wise analysis**.**

For **Human vs. Machine Detection,** we also used fine-tuned AraBERTv2 and AraModernBert for binary classification with labels are {Human, AI}. We addressed the challenge of high lexical overlap between human and machine texts by applying data augmentation by paraphrasing human samples to expand stylistic variance and make the classifier robust.

4.3 Distinguishing Configurations

LoRA vs. Full Tine-Tuning: LoRA was used only for decoder models (Gemma, Qwen) due to efficiency in large generative models. Encoder models (AraBERTv2, AraModernBert) were fully fine-tuned since they are relatively small.

Intermediate vs. Final Layers: For classification, we explicitly compared performance across layers to uncover interpretability insights using layer-wise analysis.

5 Experimental Setup

For all tasks, data was split into training, development, and test sets (70/15/15 for style transfer and author classification; 80/10/10 for human vs. machine detection), stratified by class to preserve distribution. Preprocessing included standard Arabic normalization (removing diacritics, unifying punctuation, and normalizing character variants) and model-specific tokenization with a maximum sequence length of 512. Results are summarized in Table 1.

Encoder-based models (AraBERTv2, AraModernBert) were fully fine-tuned using AdamW ($lr = 2e-5$, batch size= 4, epochs= 3, 5% warmup). Decoder-based models (Gemma, Qwen) employed LoRA adapters ($r \in \{16,32,64\}$, dropout $= 0.05$, $lr = 1e-4$), applied to attention and projection modules.

Implementation used Hugging Face Transformers (v4.41.2), PEFT (v0.11.1), PyTorch (v2.3.0), and scikit-learn (v1.5.0). Evaluation metrics varied by task: BLEU/chrF for style transfer, accuracy and macro-F1 for classification, and accuracy/F1 for machine-text detection.

Task	Split	Models	Metrics
[1]	70/15/15	Gemma, Qwen	BLEU, chrF
[2]	70/15/15	AraBERTv2, AraModernBert	Accuracy, Macro-F1
[3]	80/10/10	All	Accuracy, Macro-F1

Table 1: Experimental Setup Summary.

6 Results

In this section, we present results separately for each sub-task: Author Style Transfer (AST), Author Classification, and Human vs. Machine Detection. This structure highlights the comparative strengths of parameter-efficient fine-tuning (LoRA) and full fine-tuning across tasks.

6.1 Author Style Transfer (AST)

Table 2 reports BLEU and chrF scores for Gemma and Qwen models fine-tuned with LoRA adapters of varying ranks. The results indicate that Gemma consistently outperforms Qwen across both metrics. The best configuration is Gemma with rank r=32r=32r=32, which achieves a BLEU score of 19.04 and a chrF score of 55.14. In contrast, Qwen at rank r=16r=16r=16 performs considerably worse, obtaining a BLEU of 10.18 and chrF of 44.42.

Table 2: Results on 100 unseen Arabic articles.

Model Variant	BLEU Score	chrF Score
Gemma (r=64)	18.85	55.00
Gemma (r=32)	19.04	55.14
Gemma (r=16)	18.13	54.75
Qwen (r=16)	10.18	44.42

6.2 Author Classification

The results for author classification are presented in Table 3. AraBERTv2 achieved the highest performance, with an accuracy of 89.7% and a macro-F1 score of 0.89. AraModernBert followed with an accuracy of 87.1% and a macro-F1 score of 0.87. The layer-wise analysis provides additional insights: AraBERTv2 shows peak discriminative performance in intermediate layers (7–10), while AraModernBert encodes stylistic information more evenly across deeper layers. These findings highlight that intermediate transformer layers carry stronger stylistic signals than final layers, suggesting that representation depth plays a critical role in modeling stylistic variation in Arabic text

Table 3: Results for author classification.

Model	Accuracy	F1	Best Layer
AraBERTv2	89.71%	0.89	7
AraModernBert	87.1%	0.87	20

6.3 Human vs. Machine Detection

The binary classification results for distinguishing human- from AI-generated text are shown in Table 4. Both models reached near-ceiling performance, with AraModernBert achieving the highest accuracy of 99.4% and AraBERTv2 achieving the best macro-F1 of 0.9932.

Table 4: Results for human vs. machine detection.

Model	Accuracy	F1
AraBERTv2	99.3%	0.9932
AraModernBert	99.4%	0.9923

6.4 Comparative Insights

The comparison between full fine-tuning (for classification tasks) and LoRA (for generative tasks) highlighted clear trade-offs. Full fine-tuning enabled stable convergence and higher robustness under limited data, while LoRA delivered strong performance with fewer trainable parameters, making it attractive for scaling across multiple tasks.

To improve interpretability, we conducted a **layer-wise probing analysis**. Instead of relying only on the final [CLS] token, we extracted hidden states from each transformer layer (l = 1..12) and trained lightweight classifiers on them. Results show that **mid-level layers (7–10 in AraBERTv2)** captured the strongest stylistic and discriminative cues, while final layers tended to compress information and reduce distinctiveness. This suggests that intermediate layers preserve stylistic richness, consistent with findings in English models (Pasad et al., 2021; Van Aken et al., 2019). Figure 3 illustrates this trend for AraBERTv2 vs. AraModernBert.

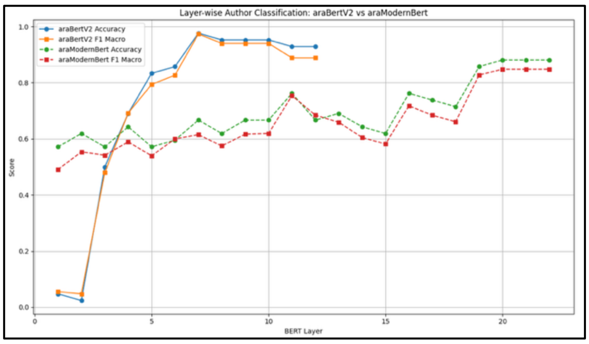


Figure 3: Layer-wise performance comparison between AraBERTv2 and AraModernBert for the author classification task. Both accuracy and

macro-F1 scores are shown across transformer layers.

Figure 3 also illustrates how performance evolves across layers of AraBERTv2 and AraModernBert. AraBERTv2 reaches peak accuracy and F1 around the middle layers (7–10), stabilizing near 0.99, while AraModernBert shows steadier gains across layers, with slightly lower but more consistent performance. This suggests AraBERTv2 encodes discriminative stylistic features earlier in its hierarchy, while AraModernBert distributes them more evenly, indicating differences in representational depth and efficiency.

6.5 Error Analysis

For author classification, common confusions occurred between authors with overlapping stylistic traits (e.g., similar sentence lengths or frequent religious expressions). For AST, errors often manifested as partial rewrites where the system retained source author lexical choices rather than fully adapting to the target style. For AI-generated text detection, misclassifications were rare but notable: in a few cases, highly fluent ChatGPT-like generations were labeled human, while noisy user-generated social media text was mislabeled as machine, showing the limits of surface-level stylistic cues.

7 Conclusion

We benchmarked Arabic LLMs on three challenging tasks including AST, author classification, and AI-generated text detection: comparing full-tuning and PEFT. Results showed that Arabic-specialized models, particularly AraBERTv2, achieve strong performance, with layer-wise analysis revealing where task-relevant features emerge. While domain sensitivity and limited benchmark resources remain challenges, this work offers one of the first multi-task evaluations of Arabic LLMs, establishing a replicable foundation and pointing toward broader dialectal coverage, cross-lingual transfer, and improved interpretability as key directions for future research.

This work highlights that PEFT, combined with careful layer-wise analysis, can unlock the full potential of Arabic LLMs, which brings stylistic shade, discriminative power, and robustness against AI-generated text detection into closer reach for underrepresented languages.

References

Abudalfa, S., Ezzini, S., Abdelali, A., Alami, H., Benlahbib, A., Chafik, S., El-Haj, M., Mahdaouy, A. El, Jarrar, M., Lamsiyah, S., & Luqman, H. (2025). The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI-Generated Text Detection. *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*.

Almanea, M. M. (2021). Automatic methods and neural networks in Arabic texts diacritization: a comprehensive survey. *IEEE Access*, *9*, 145012–145032.

Almarwani, N., & Aloufi, S. (2023). SANA at NADI 2023 shared task: Ensemble of Layer-Wise BERT-based models for Dialectal Arabic Identification. *Proceedings of ArabicNLP 2023*, 625–630.

Alqahtani, F., & Dohler, M. (2023). Survey of authorship identification tasks on Arabic texts. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(4), 1–24.

Al-Sarem, M., Saeed, F., Alsaeedi, A., Boulila, W., & Al-Hadhrami, T. (2020). Ensemble methods for instance-based Arabic language authorship attribution. *IEEE Access*, *8*, 17331–17345.

Ashqar, H. I., & Tami, M. (2025). Translation with LLMs through Prompting with Long-Form Context. *Authorea Preprints*.

Han, Z., Gao, C., Liu, J., Zhang, J., & Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *ArXiv Preprint ArXiv:2403.14608*.

Joshi, S., Khan, M. S., Dafe, A., Singh, K., Zope, V., & Jhamtani, T. (2024). Fine tuning LLMs for low resource languages. *2024 5th International Conference on Image Processing and Capsule Networks (ICIPCN)*, 511–519.

Khoboko, P. W., Marivate, V., & Sefara, J. (2025). Optimizing translation for low-resource languages: Efficient fine-tuning with custom prompt engineering in large language models. *Machine Learning with Applications*, *20*, 100649.

Liu, S., Agarwal, S., & May, J. (2024). Authorship style transfer with policy optimization. *ArXiv Preprint ArXiv:2403.08043*.

Lv, K., Yang, Y., Liu, T., Gao, Q., Guo, Q., & Qiu, X. (2023). Full parameter fine-tuning for large language models with limited resources. *ArXiv Preprint ArXiv:2306.09782*.

Masri, S., Raddad, Y., Khandaqji, F., Ashqar, H. I., & Elhenawy, M. (2024). Transformer Models in Education: Summarizing Science Textbooks with AraBART, MT5, AraT5, and mBART. *ArXiv Preprint ArXiv:2406.07692*.

Najjar, A. A., Ashqar, H. I., Darwish, O. A., & Hammad, E. (2025). Detecting AI-Generated Text in Educational Content: Leveraging Machine Learning and Explainable AI for Academic Integrity. *ArXiv Preprint ArXiv:2501.03203*.

Najjar, A., Ashqar, H. I., Darwish, O., & Hammad, E. (2025). Leveraging Explainable AI for LLM Text Attribution: Differentiating Human-Written and Multiple LLMs-Generated Text. *ArXiv Preprint ArXiv:2501.03212*.

Pasad, A., Chou, J.-C., & Livescu, K. (2021). Layer-wise analysis of a self-supervised speech representation model. *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 914–921.

Patel, A., Andrews, N., & Callison-Burch, C. (2022). Low-resource authorship style transfer: Can non-famous authors be imitated? *ArXiv Preprint ArXiv:2212.08986*.

Radhakrishnan, S., Yang, C.-H. H., Khan, S. A., Kiani, N. A., Gomez-Cabrero, D., & Tegner, J. N. (2023). A parameter-efficient learning approach to arabic dialect identification with pre-trained general-purpose speech model. *ArXiv Preprint ArXiv:2305.11244*.

Sammoudi, M., Habaybeh, A., Ashqar, H. I., & Elhenawy, M. (2024). Question-Answering (QA) Model for a Personalized Learning Assistant for Arabic Language. *ArXiv Preprint ArXiv:2406.08519*.

Tami, M., Ashqar, H. I., & Elhenawy, M. (2024). Automated Question Generation for Science Tests in Arabic Language Using NLP Techniques. *ArXiv Preprint ArXiv:2406.08520*.

Van Aken, B., Winter, B., Löser, A., & Gers, F. A. (2019). How does bert answer questions? a layer-wise analysis of transformer representations. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1823–1832.

A Appendices

For AST dataset, Figure 4 illustrates the distribution of text samples collected for various authors in a dataset used to fine-tune a LLM for Arabic author style transfer. The dataset includes prominent Arabic literary and philosophical figures, with Hassan Hanafi, Ahmad Amin, and Mohammad Hussein Heikal having the highest number of samples, indicating a richer representation of their stylistic patterns for training the model. The horizontal bars visualize the number of samples per author, supporting tasks like stylistic imitation and authorship transformation.

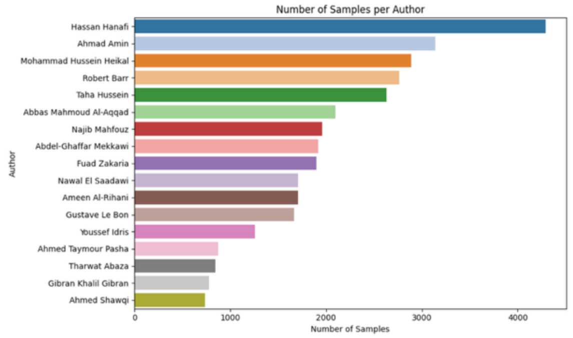


Figure 4: Number of Samples per Author in Arabic Author Style Transfer Dataset.

Moreover, Figure 5 displays the distribution of MSA text lengths, measured in number of words, across the dataset used for fine-tuning the author style transfer model. The distribution is highly concentrated around 350–400 words, with a sharp peak indicating that most samples fall within this range. The presence of a kernel density estimate (KDE) overlay highlights the unimodal and right-skewed nature of the data, where very few samples exceed 600 words. This suggests a consistent and controlled sample length throughout the dataset, which is beneficial for stable training and style learning in LLMs.

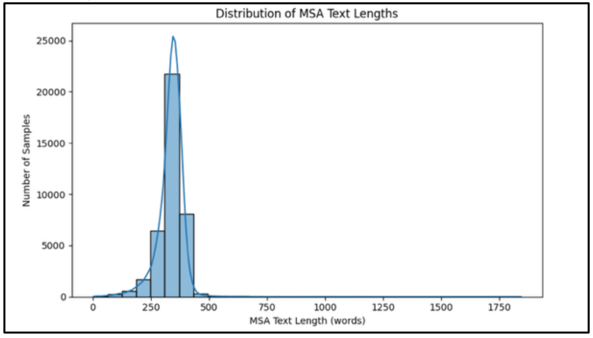


Figure 5: Distribution of MSA Text Lengths in the Arabic Author Style Transfer Dataset.

Figure 6 shows 17-word cloud subplots visualizing the most frequent and prominent words in the writings of each author from the Arabic AST dataset. The diversity of themes is evident: authors like Nawal El Saadawi and Abbas Al-Aqqad focus on gender and humanism, while Taha Hussein and Ahmad Amin emphasize thought and knowledge. Poets like Ahmed Shawqi and Gibran Khalil Gibran favor expressive and emotional lexicons, whereas philosophers such as Fuad Zakaria and Hassan Hanafi employ rational and abstract terminology.

These visualizations highlight the unique lexical fingerprints of each author, showcasing their stylistic identity. Such distinctions are foundational for fine-tuning language models to perform accurate author style transfer, as the model must learn to emulate not just surface-level vocabulary, but the deeper thematic and stylistic choices each author consistently demonstrates.

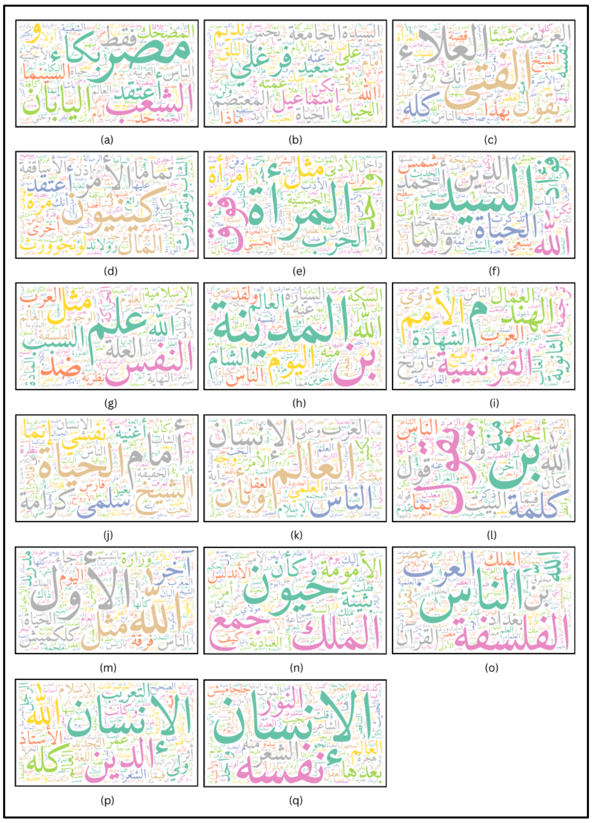


Figure 6: Word Clouds of Most Frequent Words Across 17 Arabic Authors. (a–q) show the most frequent words used by different authors in the dataset: (a) Youssef Idris, (b) Tharwat Abaza, (c) Taha Hussein, (d) Robert Barr, (e) Nawal El Saadawi, (f) Najib Mahfouz, (g) Hassan Hanafi, (h) Mohammad Hussein Heikal, (i) Gustave Le Bon, (j) Gibran Khalil Gibran, (k) Fuad Zakaria, (l) Ahmed Taymour Pasha, (m) Ameen Al-Rihani, (n) Ahmed Shawqi, (o) Ahmad Amin, (p) Abbas Mahmoud Al-Aqqad, and (q) Abdel-Ghaffar

Mekkawi. Each subplot highlights the author's dominant vocabulary, providing insight into their unique lexical and thematic style.

For Author Classification, the t-SNE visualization shown in Figure 7 represents the clustering of Arabic text samples based on [CLS] token embeddings produced by a fine-tuned AraBERTv2 model, trained for the task of author classification. Each point represents a text sample, and colors correspond to different authors. The embeddings were projected into 2D space using t-SNE for visualization purposes.

Figure 7 illustrates how well the fine-tuned AraBERTv2 model captures the distinct stylistic and semantic features of different authors in the dataset. Clear and well-separated clusters, such as those for Nawal El Saadawi, Taha Hussein, and Robert Barr, suggest that the model has successfully learned author-specific linguistic patterns, enabling high confidence in distinguishing between them.

Some clusters are positioned close to others (e.g., Ahmad Amin and Mohammad Hussein Heikal), indicating potential stylistic or thematic similarities between those authors' writing. Meanwhile, others like William Shakespeare (likely translated texts) or George Zaidan show strong separation, hinting at distinct lexical or structural traits.

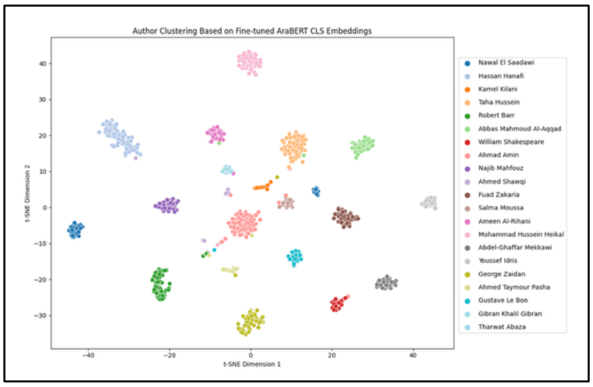


Figure 7: Author Clustering Based on Fine-Tuned AraBERT CLS Embeddings.

AraHealthQA 2025: The First Shared Task on Arabic Health Question Answering

Hassan Alhuzali,[1] Walid Al-Eisawi ,[2] Muhammad Abdul-Mageed,[3] Chaimae Abouzahir[2] Mouath Abu-Daoud,[2] Ashwag Alasmari,[4] Renad Al-Monef,[4] Ali Alqahtani,[4] Lama Ayash,[4] Leen Kharouf,[2] Farah E. Shamout,[2] Nizar Habash[2]

[1]Umm Al-Qura University, [2]New York University Abu Dhabi,
[3]The University of British Columbia, [4]King Khalid University

Correspondence: hrhuzali@uqu.edu.sa & farah.shamout@nyu.edu

Abstract

We introduce AraHealthQA 2025, the Comprehensive Arabic Health Question Answering Shared Task, held in conjunction with ArabicNLP 2025 (co-located with EMNLP 2025). This shared task addresses the paucity of high-quality Arabic medical QA resources by offering two complementary tracks: MentalQA, focusing on Arabic mental health Q&A (e.g., anxiety, depression, stigma reduction), and MedArabiQ, covering broader medical domains such as internal medicine, pediatrics, and clinical decision making. Each track comprises multiple subtasks, evaluation datasets, and standardized metrics, facilitating fair benchmarking. The task was structured to promote modeling under realistic, multilingual, and culturally nuanced healthcare contexts. We outline the dataset creation, task design and evaluation framework, participation statistics, baseline systems, and summarize the overall outcomes. We conclude with reflections on the performance trends observed and prospects for future iterations in Arabic health QA[1].

1 Introduction

Large Language Models (LLMs) have demonstrated substantial potential across a wide range of healthcare applications, including clinical decision support, patient triage, and automated question answering. Despite this progress, their effectiveness in the Arabic medical domain remains largely underexplored, mainly due to a lack of high-quality, domain-specific datasets and standardized benchmarking efforts. Existing resources for Arabic healthcare are limited in size, coverage, and linguistic diversity, particularly for mental health, which presents unique challenges related to cultural context, language variation, and sensitive content.

To address these limitations, AraHealthQA 2025 introduces a new shared task aimed at evaluating and advancing the performance of LLMs on Arabic medical question-answering tasks. The shared task provides carefully curated datasets covering both general health and mental health inquiries, along with clearly defined subtasks for classification and answer generation. By establishing a structured evaluation framework, AraHealthQA 2025 enables systematic benchmarking of models, encourages reproducible research, and fosters the development of LLMs that can provide accurate, contextually aware, and culturally sensitive responses in realistic healthcare scenarios.

AraHealthQA 2025 consists of two complementary tracks, each targeting a distinct area of Arabic healthcare question answering. Figure 1 shows an overview of the AraHealthQA 2025 Shared Task. The first track, Arabic Mental Health QA (MentalQA), focuses on mental health topics including anxiety, depression, cognitive disorders, therapeutic practices, and stigma reduction. This track is designed to evaluate models across three subtasks: question classification, answer classification, and question answering. The dataset for this track includes 500 question-answer pair, enabling participants to build models capable of understanding diverse question types, answer strategies, and generating contextually appropriate responses. This track emphasizes the importance of culturally aware and clinically relevant NLP systems in the Arabic mental health context.

The second track, General Arabic Health QA (MedArabiQ), addresses a broader spectrum of medical domains, such as internal medicine, cardiology, pediatrics, and medical education. It includes two subtasks: multiple-choice question answering and open-ended question answering. This track allows evaluation of models on both structured and open-ended formats, assessing their ability to provide accurate, relevant, and well-formed responses across general medical knowledge.

By providing these two tracks, AraHealthQA

[1] Author order, excluding the first two lead authors, is alphabetical. The final author served in an advisory role.

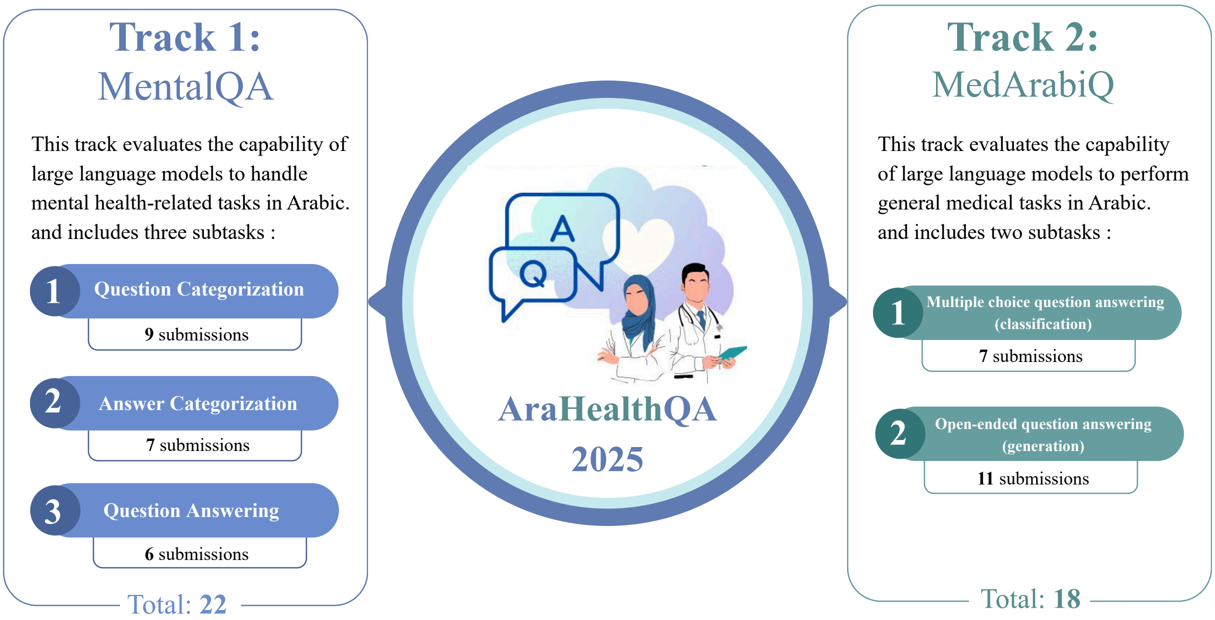

Track 1:
MentalQA
This track evaluates the capability of large language models to handle mental health-related tasks in Arabic. and includes three subtasks :
1 Question Categorization
9 submissions
2 Answer Categorization
7 submissions
3 Question Answering
6 submissions
Total: 22
A
Q
AraHealthQA
2025
Track 2:
MedArabiQ
This track evaluates the capability of large language models to perform general medical tasks in Arabic. and includes two subtasks :
1 Multiple choice question answering (classification)
7 submissions
2 Open-ended question answering (generation)
11 submissions
Total: 18

Figure 1: An Overview of the AraHealthQA 2025 Shared Task.

2025 aims to create a comprehensive evaluation framework for Arabic healthcare NLP. Participants are encouraged to develop systems that not only perform well in classification or generation tasks but also demonstrate cultural and domain awareness, supporting practical and research applications in both mental health and general medical contexts.

2 Related Work

Research on mental health and medical NLP has gained significant interest in recent years, with particular attention given to the creation of specialized datasets and benchmarks. However, most of these efforts have been concentrated on English, leaving Arabic largely underexplored despite its wide usage and the pressing healthcare needs of Arabic-speaking populations. This section reviews prior work relevant to the two tracks of our shared task: MentalQA and MedArabiQ.

2.1 Mental Health Benchmarks

Existing mental health studies have largely focused on specific disorders, including suicidal attempts, self-injury, loneliness, depression, and anxiety, which can limit the generalizability of AI models across broader mental health issues (Shen et al., 2017; Turcan and Mckeown, 2019; Rastogi et al., 2022; Garg et al., 2023). More specialized resources capture emotions associated with particular conditions: the CEASE dataset (Ghosh et al., 2020) targets emotions of suicide attempters, while EmoMent (Atapattu et al., 2022) focuses on emotional states linked to depression and anxiety. Other datasets support tasks such as identifying pain levels in mental health notes (Chaturvedi et al., 2023) or extracting causal interpretations from clinical narratives, as in CAMS (Garg et al., 2022).

Despite these global efforts (Atapattu et al., 2022; Kabir et al., 2022; Sun et al., 2021; Alasmari et al., 2023; Ghosh et al., 2020; Chaturvedi et al., 2023; Garg et al., 2022; Alasmari, 2025), Arabic remains an understudied language in mental health NLP. Only a few studies have addressed mental health tasks in Arabic texts (Aldhafer and Yakhlef, 2022; Al-Musallam and Al-Abdullatif, 2022; Al-Laith and Alenezi, 2021). For example, Aldhafer and Yakhlef (Aldhafer and Yakhlef, 2022) developed depression detection models from Arabic tweets, accounting for cultural stigma, while Al-Musallam and Al-Abdullatif (Al-Musallam and Al-Abdullatif, 2022) applied feature-based machine learning techniques for depression detection in Arabic texts.

To bridge this gap, the MentalQA dataset (Alhuzali and Alasmari, 2025; Alhuzali et al., 2024) was developed, providing annotated Arabic question-answer pairs that cover a variety of question types and answer strategies. This dataset supports the creation and evaluation of NLP systems capable of handling various mental health inquiries, forming the foundation of the Arabic Mental Health Question Answering Shared Task.

Using the MentalQA dataset, this track provides a dedicated benchmark for Arabic mental health question-answering. It addresses the dual challenge of classification and response generation, creating a platform to systematically evaluate models in a culturally sensitive setting. Through this effort, MentalQA promotes research on the building of reliable and context-sensitive NLP systems for Arabic-speaking communities.

2.2 General Health Benchmarks

The evaluation of LLMs for medical applications has been dominated by English-centric sources and, typically, exam-style question-answering datasets. The Massive Multitask Language Understanding suite (MMLU) includes a subset derived from the USMLE (Hendrycks et al., 2021). Similarly, MedQA assesses board-exam-style QA and broadens multilingual coverage by incorporating traditional and simplified Chinese alongside English (Jin et al., 2020). Building on these efforts, Gao et al. (2023) introduce Dr. Bench, an English-only diagnostic reasoning benchmark in clinical NLP that targets understanding of clinical narratives, medical knowledge reasoning, and the generation of differential diagnoses.

In contrast, Arabic medical evaluation resources remain comparatively scarce and unevenly distributed across tasks. Notable efforts include AraSTEM, which targets question answering with a medical subset (Mustapha et al., 2024), and AraMed, which provides an Arabic medical corpus paired with an annotated QA dataset (Alasmari et al., 2024). A translation-based dataset also exists, wherein Achiam et al. (2023) converted MMLU into 14 languages, including Arabic. While valuable, these resources still leave substantial portions of the Arabic medical task space unattended, highlighting the need for dedicated benchmarking.

With the vast potential of LLMs in healthcare, it is crucial to accommodate Arabic-speaking patients to ensure fair deployment. This motivated the development of the MedArabiQ benchmark (Daoud et al., 2025) for Arabic medical tasks, upon which this track of the shared task is based. The benchmark covers medical education and patient-clinician conversation in Arabic, with initial results indicating generally poor performance of LLMs on these tasks. This prompted us to introduce this shared task, inviting researchers to enhance models' capabilities in the Arabic medical task domain.

3 Task Overview

3.1 Track 1: MentalQA

The objective of Track 1 is to assess the capabilities of LLMs in addressing healthcare-related tasks in Arabic, with a particular emphasis on the mental health domain. Given the sensitivity and cultural nuances of mental health conversations, this track aimed to benchmark models on their ability to classify questions, identify appropriate answer strategies, and generate supportive, contextually relevant responses in Arabic. This track was built upon the MentalQA dataset, the first publicly available annotated Arabic dataset for mental health support.

The dataset covers a variety of question types (e.g., diagnosis, treatment, anatomy & physiology, epidemiology, healthy lifestyle, provider choices, or other) and answer strategies (information provision, direct guidance, and emotional support), and is based on real patient inquiries paired with expert doctor responses for question-answering. Participants competed in three subtasks, each targeting a different aspect of mental health NLP systems. We now turn to a detailed description of each subtask, including objectives, dataset splits, and evaluation.

3.1.1 Subtask 1 and 2

We propose Subtask 1: Question Type Classification and Subtask 2: Answer Strategy Classification, which share a similar multi-label classification setup. In Subtask 1, systems must classify each user question into one of several predefined types. In Subtask 2, systems must predict the answer strategy employed in a response, noting that multiple strategies may co-occur.

For both subtasks, the dataset is based on MentalQA and is divided into 300 samples for training, 50 samples for development, and 150 samples as a blind test set for final evaluation. The training set can be used to fine-tune LLMs or serve as a base for few-shot learning approaches. The development set is intended to tune hyper-parameters and evaluate performance, while the test set ensures fair benchmarking of all participants.

3.1.2 Subtask 3

We propose Subtask 3: Question Answering, where systems are required to generate concise, supportive, and contextually appropriate answers in Arabic. This task forms the basis for a robust question-answering system capable of providing specialized responses to a wide range of mental health-related inquiries. The dataset is also based on MentalQA (Alhuzali and Alasmari, 2025; Alhuzali et al., 2024) and follows the same split described in Subtask 1 and 2.

3.2 Track 2: MedArabiQ

The objective of this track was to evaluate the capabilities of LLMs in performing healthcare-related tasks in Arabic, across a variety of general medical domains. The track consists of two subtasks that reflect critical scenarios in clinical education and practice, aiming to benchmark both classification and generative performance in realistic medical settings.

The development set was provided as the entire original MedArabiQ dataset (Daoud et al., 2025), consisting of 700 multiple-choice and open-ended questions, whereas the test set consisted of similar but entirely new, unseen questions. The order of questions was entirely random.

3.2.1 Subtask 1

The first subtask focuses on multiple-choice question answering as a classification task, with questions that include standard multiple-choice, multiple-choice questions with potentially biased distractors, and fill-in-the-blank questions with a set of candidate answers. The objective is to assess the model's ability to apply clinical knowledge in structured decision-making scenarios. The dataset provided to candidates consisted of a development set of 300 samples, which can be used for model training and validation, and a blind test set of 100 samples.

The test set for Subtask 1 consisted of 50 multiple-choice questions and 50 fill-in-the-blank questions with choices. Initially, 100 multiple-choice questions were randomly sampled from a larger repository of questions from past regional Arabic medical exams. These questions were digitized and extracted from physical exam papers, eliminating any risk of contamination. Then, 50 of these multiple-choice questions were converted into fill-in-the-blank questions, following the methodology of previous work (Daoud et al., 2025). By randomly sampling from the same source, a similar distribution of medical specialties and difficulty levels was retained. Additionally, 74% of questions provided four answer choices, whereas the remaining 26% offered five.

3.2.2 Subtask 2

The second subtask presents fill-in-the-blank and open-ended question answering as a generative task. Participants were tasked with generating free-text responses to prompts that include questions without predefined options. The goal in this track is to evaluate model responses for semantic alignment with the reference answers, either from clinicians or textbook ground truth answers. The dataset for this subtask consisted of a development set of 400 samples, which can be used for training and validation, and a blind test set of 100 samples.

The test set included 50 fill-in-the-blank questions without choices–constructed from randomly sampled multiple-choice questions, similar to Subtask 1–as well as 50 patient-doctor questions. The patient-doctor questions were randomly sampled from AraMed (Alasmari et al., 2024), which is also used as a source for MedArabiQ (Daoud et al., 2025).

4 Shared Task Teams

Submission Rules: For Track 1, we allowed participant teams to submit up to five runs for each test set and for each of the three subtasks. For Track 2, participants were initially allowed 10 submissions each, which was later increased to 15 submissions due to platform-specific issues. For each team, only the submission with the highest score was retained on the official leaderboard. The official evaluation relied on a blind test set. To ensure fairness and reproducibility, each subtask of each track was hosted as a separate competition on Codabench (Xu et al., 2022), enabling automatic scoring and ranking of submissions. These Codabench instances will remain active even after the official competition concludes, supporting continued experimentation and benchmarking on the MentalQA and MedArabiQ datasets.

4.1 Track 1: MentalQA

Evaluation: Subtask 1 and Subtask 2 are multi-label classification tasks and are evaluated using Weighted F1 score and Jaccard score. The Weighted F1 balances precision and recall while accounting for class imbalance, whereas the Jaccard score measures the overlap between predicted and gold label sets, making it suitable for multi-label evaluation. Subtask 3 is evaluated using BERTScore (Zhang et al., 2020), which leverages contextual embeddings from pre-trained language models to capture semantic similarity between generated responses and reference answers. Together, these metrics provide a robust assessment of system performance across the classification and generation subtasks, reflecting both the accuracy of the labels and the semantic quality of the outputs.

Participating Teams: A total of 46 unique teams registered for the shared task. During the testing phase, teams were allowed up to five submissions each. The breakdown across the subtasks is as follows: 9 submissions for Subtask 1 from 9 unique teams, 7 submissions for Subtask 2 from 7 unique teams, and 6 submissions for Subtask 3 from 6 unique teams. We received ten description papers, all of which were accepted for publication as presented in Table 1.

Baselines: For Subtask 1 and Subtask 2, we employed a simple yet strong baseline based on the most frequent label strategy. In this setting, the model always predicts the most common category (or set of categories) observed in the training data, regardless of the input. Although this baseline does not leverage the semantic content of the questions or answers, it provides a meaningful lower bound for performance and highlights the inherent class imbalance in the dataset. This baseline is commonly used in shared tasks to establish a reference point against which more sophisticated approaches can be fairly compared.

4.2 Track 2: MedArabiQ

Evaluation: For Subtask 1, we used accuracy as the evaluation metric, given that it is a classification task. Since Subtask 2 is a generation task, submissions were evaluated against the ground truth answers using BERTScore to capture semantic similarity between the two texts.

Participating Teams: A total of 26 participants registered across both subtasks, including seven who submitted predictions for Subtask 1 and eleven who submitted for Subtask 2. System description papers were received from a total of five teams, including three for Subtask 1 and five for Subtask 2. A summary of participating teams is provided in Table 1.

Baselines: For Subtask 1, we chose to use both Gemini 1.5 Pro (Georgiev et al., 2024) and DeepSeek v3 (DeepSeek-AI et al., 2025) as baselines, based on existing results that show that Gemini achieves the highest accuracy on multiple-choice questions, while DeepSeek performs the strongest on fill-in-the-blank questions with choices (Daoud et al., 2025). Since our test set includes both types of questions, we compare results to both to ensure a strong, realistic baseline. For Subtask 2, we only used Gemini 1.5 Pro as our baseline, seeing as it achieved the highest BERTScore on fill-in-the-blank questions without choices and performed comparably to other models on patient-doctor Q&A. The prompts used for evaluating baseline models were constructed based on similar literature (Daoud et al., 2025).

5 Results

5.1 Track 1: MentalQA

5.1.1 Subtask 1

The results of Subtask 1 shown in Table 2 reveal a range of performances among participating teams, with Weighted-F1 scores spanning from 0.61 to 0.24 as presented in Table 2. The top-performing system, *mucAI*, achieved a Weighted-F1 of 0.61 and a Jaccard score of 0.53, closely followed by *Binary_Bunch* with nearly identical results. At the lower end, the baseline model obtained the weakest performance, with a Weighted-F1 of 0.24 despite a relatively higher Jaccard score of 0.40. This indicates that frequency-based methods were insufficient for handling the task effectively, while most submitted systems provided substantial improvements over the baseline.

A closer comparison highlights several interesting patterns. While *mucAI* and *Binary_Bunch* led the rankings, other teams such as *Sindbad* and *Quasar* achieved relatively balanced performance across both metrics, suggesting more consistent predictions. In contrast, *Fahmni* attained a lower Weighted-F1 of 0.44 yet a relatively competitive Jaccard score of 0.45, pointing to broader label coverage but reduced precision. Moreover, *RetAug* and *AraMinds* produced identical scores, implying comparable modeling strategies or effectiveness. These results collectively illustrate the diversity in system behaviors and the varying trade-offs between precision and recall across participating teams.

5.1.2 Subtask 2

The results of Subtask 2 presented in Table 3 show overall stronger performance compared to Subtask 1, with Weighted-F1 scores ranging from 0.79 to 0.44 as shown in Table 3. The top-performing teams, *Sindbad* and *MarsadLab*, both achieved the highest Weighted-F1 score of 0.79, while *Binary_Bunch*, *AraMinds*, and *Quasar* followed

Team	Affiliation	Tasks
	Track 1: MentalQA	
mucAI (Abdou, 2025)	-	1,2
Binary_Bunch (Bhattacharjee et al., 2025)	Chittagong University, Bangladesh	1, 2
MarsadLab (Bessghaier et al., 2025)	Hamad Bin Khalifa University, Qatar; Northwestern University, Qatar	1, 2
Sindbad (Morsy et al., 2025)	George Washington University, USA	1, 2, 3
Quasar (Chowdhury and Chowdhury, 2025)	Chittagong University, Bangladesh	1, 2
RetAug (AbdelAziz et al., 2025)	Nile University, Egypt	1,2, 3
AraMinds (Zaytoon et al., 2025)	Alexandria University, Egypt	1, 2, 3
Fahmni (Sabty et al., 2025)	MBZUAI, UAE; Gameball Company; German International University, Egypt; American University in Cairo, Egypt	1, 2, 3
Sakinah-AI (Elden and Abukar, 2025)	Cairo University, Egypt; University of South Wales, UK	1
MindLLM (Eshaq, 2025)	King Khalid University, Saudi Arabia	3
	Track 2: MedArabiQ	
!MSA (Tarek et al., 2025)	MSA University, Egypt	1, 2
MedLingua (Emad Eldin and Abukar, 2025)	Cairo University, Egypt; University of South Wales, UK	1, 2
NYUAD (AlDahoul and Zaki, 2025)	New York University Abu Dhabi	1, 2
MedGapGab (Hikal, 2025)	University of Göttingen, Germany	2
Egyhealth (Amer et al., 2025)	Nile University, Egypt	2

Table 1: List of teams that participated in Track 1 and Track 2 of AraHealthQA 2025.

Team	Weighted-F1	Jaccard Score
mucAI	0.61	0.53
Binary_Bunch	0.60	0.53
MarsadLab	0.55	0.41
Sindbad	0.53	0.49
Quasar	0.52	0.41
RetAug	0.49	0.28
AraMinds	0.49	0.28
Fahmni	0.44	0.45
Sakinah-AI	0.34	0.20
Baseline (MF)	0.24	0.40

Table 2: Performance of the systems on the test set of **Subtask 1 of Track 1**. Results are sorted by Weighted F1 score.

Team	Weighted-F1	Jaccard Score
Sindbad	0.79	0.71
MarsadLab	0.79	0.67
Binary_Bunch	0.77	0.71
AraMinds	0.76	0.68
Quasar	0.76	0.66
Fahmni	0.69	0.62
Baseline (MF)	0.44	0.56

Table 3: Performance of the systems on the test set of **Subtask 2 of Track 1**. Results are sorted by Weighted F1 score.

closely with scores between 0.76 and 0.77. At the lower end, the baseline system attained a Weighted-F1 of 0.44, which is notably weaker than all submitted systems, although its Jaccard score of 0.56 was higher than that of some teams, reflecting a bias toward broader label prediction coverage.

A comparative analysis highlights several important trends. While *Sindbad* and *Binary_Bunch* obtained identical Jaccard scores of 0.71, suggesting strong recall and balanced predictions, *MarsadLab* matched the top Weighted-F1 but with a slightly lower Jaccard score of 0.67, indicating stronger precision but somewhat reduced coverage. Similarly, *Fahmni* scored considerably lower on Weighted-F1 (0.69) but still maintained a competitive Jaccard score of 0.62, suggesting that it captured a broader set of relevant labels despite less precise predictions. These results highlight the close competition among top systems and the subtle variations in the precision–recall balance across teams.

5.1.3 Subtask 3

The results of Subtask 3 depicted in Table 4, evaluated using BERTScore, demonstrate a narrower performance range compared to the earlier subtasks, with scores spanning from 0.679 to 0.646 as illustrated in Table 4. The best-performing system, *RetAug*, achieved a BERTScore of 0.679, closely followed by *MindLLM* and *Sindbad* with scores of 0.670 and 0.668, respectively. The remaining teams, including *AraMinds*, *MarsadLab*, and *Fahmni*, all produced scores above 0.64, indicating that even the lowest-performing system performed reasonably well within a relatively tight margin.

In contrast to Subtasks 1 and 2, where the differences between the top and bottom systems were more pronounced, the small performance gap in Subtask 3 highlights the increased difficulty of the task and the challenge of distinguishing system

Team	BERTScore
RetAug	0.679
MindLLM	0.670
Sindbad	0.668
AraMinds	0.663
Fahmni	0.646

Table 4: Performance of the systems on the test set of **Subtask 3 of Track 1**.

quality using automatic evaluation alone. We observed that models often struggled with generating culturally sensitive and context-appropriate responses, despite achieving relatively high overlap-based scores. This suggests that automatic metrics such as BERTScore, while useful, may not fully capture the nuances required to evaluate responses in the mental health domain.

5.1.4 General Description of Submitted Systems (Track 1)

The following provides an overview of the leading systems submitted to the AraHealthQA 2025 MentalQA Track 1. Each subtask highlights the winning team, their methodology, and the core strategies that enabled high performance.

Subtask 1: The winning team, **mucAI** (Abdou, 2025), achieved a weighted F1-score of 0.61 for question classification. Their system, *Explain–Retrieve–Verify (ERV)*, is a lightweight, training-free pipeline for multi-label categorization of Arabic mental-health questions. ERV combines a chain-of-thought LLM classifier with example-based retrieval and a verification agent. The LLM proposes candidate labels and rationales, a similarity agent retrieves top-k nearest questions via multilingual sentence-transformer embeddings to provide case-based priors, and the verification agent reconciles these signals to produce a final label set with calibrated confidence. A post-processing step handles code parsing and confidence clamping. ERV runs efficiently at inference time without requiring fine-tuning or external data.

Subtask 2: The winning team, **Sindbad** (Morsy et al., 2025), achieved a weighted F1-score of 0.71 and a Jaccard score of 0.71 for answer classification. Their approach leverages dataset augmentation to balance underrepresented classes, followed by a rigorous pipeline that uses state-of-the-art pre-trained language models (PLMs) and large language models (LLMs) for few-shot prompting and instruction fine-tuning. They utilize Gradient-free Edit-based Instruction Search (GrIPS) to optimize prompt selection, improving the quality and consistency of the QA system without extensive manual intervention.

Subtask 3: The winning team, **RetAug** (AbdelAziz et al., 2025), achieved a BERTScore of 0.679 for generative question answering. Their system employs a Retrieval-Augmented Generation (RAG) framework tailored for Arabic mental health Q&A. User queries are normalized and enhanced to handle dialectal variations, then matched with relevant contexts through hybrid retrieval, combining dense embeddings (Arabic-SBERT-100K) and sparse BM25 search. Retrieved contexts are reranked using semantic similarity, BM25 score, text length, and question similarity, with culturally sensitive filtering to ensure safe and appropriate advice. Finally, a fine-tuned `Saka-14B` model generates responses using prompts that integrate the user query, top contexts, domain-specific instructions, and cultural constraints. This approach allows RetAug to produce contextually relevant and culturally appropriate answers while effectively grounding the generation in retrieved knowledge.

5.2 Track 2: MedArabiQ

5.2.1 Subtask 1

With three teams participating in Subtask 1, the results shown in Table 5 fall within a close range. The strongest performing team, *NYUAD*, achieved an accuracy of 0.77, while the weakest system was still a relatively impressive accuracy of 0.74, achieved by *MedLingua*. At second place, *!MSA* achieved a similar accuracy of 0.76. The lack of variance in results can be attributed to the small sample size, as well as similarities in approaches. All three teams significantly outperform both baselines, Gemini and DeepSeek.

5.2.2 Subtask 2

Despite the fact that more submissions were received for Subtask 2, there was even less variance observed in the results, as seen in Table 6. While the strongest team, *MedGapGab*, achieved a BERTScore of 0.873, it only outperformed the second strongest team, *!MSA*, by a margin of 0.003, and the weakest team, *MedLingua*, by a margin of 0.011. The third and fourth-highest performing teams, respectively, were NYUAD and Egyhealth, achieving BERTScores of 0.864 and 0.863. These all appear to indicate strong performance in open-ended Arabic medical tasks, outperforming the Gemini baseline, which achieves a slighly

Team	Accuracy
NYUAD	0.77
!MSA	0.76
MedLingua	0.74
Gemini 1.5 Pro	0.47
DeepSeek v3	0.51

Table 5: Performance of the systems on the test set of **Subtask 1 of Track 2**. Results are sorted by accuracy. Gemini 1.5 Pro and DeepSeek v3 are included as baselines.

Team	BERTScore
MedGapGab	0.873
!MSA	0.870
NYUAD	0.864
Egyhealth	0.863
MedLingua	0.862
Gemini 1.5 Pro	0.844

Table 6: Performance of the systems on the test set of **Subtask 2 of Track 2**. Results are sorted by BERTScore. The performance of Gemini 1.5 Pro is included as a baseline

lower BERTScore of 0.844.

5.2.3 General Description of Submitted Systems (Track 2)

The following provides an overview of the leading systems submitted to the AraHealthQA 2025 MentalQA Track 2. Each subtask highlights the winning team, their methodology, and the core strategies that enabled high performance.

Subtask 1: The winning team, **NYUAD**, achieved an accuracy of 0.77. AlDahoul and Zaki (2025) employed a multifaceted approach, evaluating numerous proprietary base LLMs including several models from Gemini, DeepSeek, GPT (Achiam et al., 2023), and Llama (Grattafiori et al., 2024). Their findings revealed that Gemini Pro 2.5 achieved the strongest performance at an accuracy of 0.76, followed by Gemini Flash 2.5 and GPT-o3 at 0.74. Prompt engineering and chain-of-thought (CoT) reasoning were prominent factors in their success, as they constructed a detailed zero-shot prompt in Arabic that instructed the model to think step-by-step, explain relevant concepts, pinpoint incorrect options, and refer to reputable medical facts to arrive at an answer. This outperformed a simple English-language prompt, which did not involve CoT or any notable prompt engineering. To further improve the accuracy of their system, AlDahoul and Zaki (2025) employed a majority voting technique using predictions from the three top-performing base LLMs.

Subtask 2: The team that submitted the highest-performing system was **MedGapGab**, which achieved a BERTScore of 0.873. Hikal (2025) developed a modular, model-agnostic system that addressed the different subtypes of questions, specifically fill-in-the-blank questions and patient-doctor Q&A. For each question, the approach involved initially classifying the question into either category, before using Term Frequency-Inverse Document Frequency (TF-IDF) to retrieve the four most similar examples from the development set. These would then be inserted into a task-specific prompt, providing detailed context and specific, informative instructions to the model. Finally, each question was routed to either Gemini 2.5 Flash or DeepSeek V3. With the former optimized for precise terminology and the latter optimized for reasoning, the system exploits the strengths of each model to complete different tasks. The modularity of this system is instrumental in its success in the shared task.

6 Discussion and Conclusion

The AraHealthQA 2025 shared task represents a significant step toward advancing Arabic healthcare NLP, particularly in the underexplored domains of mental health dialogue and medical question answering. Insights from both tracks highlight recurring challenges and opportunities for progress. A key finding is the critical role of domain-specific resources. While large multilingual LLMs have shown strong performance in general contexts, many systems struggled to generate accurate and culturally appropriate responses for Arabic healthcare, especially in mental health. This reinforces the importance of curated benchmarks such as MentalQA and MedArabiQ, which enable models to address sensitive topics like depression, stigma, and medical reasoning with greater nuance.

Differences in modeling strategies further revealed clear trends. Teams variously employed multilingual or Arabic-specific pretrained models alongside prompt engineering, instruction tuning, and parameter-efficient fine-tuning. Systems that blended domain adaptation with lightweight fine-tuning generally outperformed zero-shot prompting baselines, underscoring the value of hybrid approaches that combine foundation model strengths with healthcare-specific knowledge. Prompt design emerged as consistently effective across tracks, though interestingly zero-shot prompting sometimes surpassed few-shot setups, suggesting irrelevant examples can trigger hallucinations. Similarities in approaches employing test-time techniques, combined with relatively small dataset size, resulted in little variance in results for Track 2.

Evaluation outcomes also highlighted task-specific trade-offs. Teams achieved stronger results in structured subtasks (e.g., multi-label classification) than in open-ended QA, where correctness must be balanced with empathy and cultural sensitivity. While automatic metrics such as BERTScore captured surface-level alignment, they failed to fully measure appropriateness or trustworthiness, pointing to the necessity of human-in-the-loop evaluation, particularly with clinicians and native speakers. Despite constraints in Track 2, such as restrictions on fine-tuning with task data and limited availability of Arabic medical resources, teams demonstrated that careful prompt design, in-context learning, and ensemble methods can substantially improve over baselines. Nevertheless, progress in Arabic healthcare NLP will require not only richer datasets but also stronger collaborations between NLP researchers, clinicians, and mental health professionals to ensure that future systems are accurate, culturally aware, and ethically aligned.

Looking ahead, future iterations of AraHealthQA aim to expand both scale and scope. Planned directions include releasing larger and more diverse datasets, extending coverage to additional medical specialties, and incorporating multilingual benchmarks to reflect the linguistic diversity of healthcare in the Arab world. Human-in-the-loop evaluations with domain experts will be a key priority to ensure clinical reliability. Through these efforts, AraHealthQA seeks to catalyze sustained research at the intersection of Arabic NLP, healthcare, and AI for social good.

7 Limitations and Ethical Considerations

While this shared task provides an important step toward advancing Arabic NLP for healthcare applications, several limitations should be acknowledged. First, the datasets used in both tracks are constrained in size compared to English counterparts, which may restrict model generalizability and lead to overfitting. Furthermore, the focus on Arabic mental health and medical texts, though novel, does not yet capture the full diversity of dialects, socio-cultural contexts, or clinical domains within the Arabic-speaking world. This highlights the need for larger, more representative, and multi-dialectal datasets in future iterations.

From an ethical perspective, the sensitive nature of healthcare and mental health data raises significant concerns. Although the MentalQA and MedArabiQ datasets were curated from publicly available or anonymized sources, there remains a risk of models generating misleading, unsafe, or culturally inappropriate responses. Deploying such systems in real-world clinical or mental health settings without rigorous human oversight could result in harm to patients. Therefore, outputs from participating systems should be regarded strictly as research artifacts rather than clinical advice.

We also recognize the ethical imperative of ensuring inclusivity and fairness. Biases present in training data may propagate into model predictions, potentially amplifying stigma or misrepresenting vulnerable groups. To mitigate these risks, future efforts should include robust bias evaluation, collaboration with domain experts, and incorporation of human-in-the-loop approaches. By doing so, the shared task can contribute not only to advancing NLP research but also to supporting equitable and responsible healthcare technologies.

Acknowledgments

We thank the organizers, contributors, annotators, reviewers, and sponsors of AraHealthQA 2025.

References

Abdelaziz Amr AbdelAziz, Mohamed Ahmed Youssef, Mamdouh Mohamed Koritam, Marwa ELDeeb, and Ensaf Hussein. 2025. Retaug at arabic mental health question answering: A multi-task approach with advanced retrieval-augmented generation. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1.*

Ahmed Abdou. 2025. mucai at arahealthqa 2025: Explain-retrieve-verify (erv) workflow for multi-label arabic health qa classification. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1.*

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ali Al-Laith and Mamdouh Alenezi. 2021. Monitoring people's emotions and symptoms from arabic tweets during the COVID-19 pandemic. *Information (Basel)*, 12(2):86.

N Al-Musallam and M Al-Abdullatif. 2022. Depression detection through identifying depressive arabic tweets from saudi arabia: Machine learning approach. In *2022 Fifth National Conference of Saudi Computers Colleges (NCCC)*, pages 11–18.

Ashwag Alasmari. 2025. A scoping review of arabic natural language processing for mental health. In *Healthcare*, volume 13-9, page 963. MDPI.

Ashwag Alasmari, Sarah Alhumoud, and Waad Alshammari. 2024. Aramed: Arabic medical question answering using pretrained transformer language models. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, pages 50–56.

Ashwag Alasmari, Luke Kudryashov, Shweta Yadav, Heera Lee, and Dina Demner-Fushman. 2023. Chq-socioemo: Identifying social and emotional support needs in consumer-health questions. *Scientific Data*, 10(1):329.

Nouar AlDahoul and Yasir Zaki. 2025. Nyuad at arahealthqa shared task: Benchmarking the medical understanding and reasoning of large language models in arabic healthcare tasks. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 2*.

Shahad Hathal Aldhafer and Mourad Yakhlef. 2022. Depression detection in arabic tweets using deep learning. In *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. IEEE.

Hassan Alhuzali and Ashwag Alasmari. 2025. Pretrained language models for mental health: An empirical study on arabic q&a classification. *Healthcare*, 13(9).

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*, 12:101155–101165.

Hossam Amer, Rawan Tarek Taha, Gannat Ibrahim, and Ensaf Hussein Mohammed. 2025. Egyhealth at general arabic health qa (medarabiq): An enhanced rag framework with large scale arabic q&a medical data. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 2*.

Thushari Atapattu, Mahen Herath, Charitha Elvitigala, Piyanjali de Zoysa, Kasun Gunawardana, Menasha Thilakaratne, Kasun de Zoysa, and Katrina Falkner. 2022. Emoment: An emotion annotated mental health corpus from two south asian countries. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6991–7001.

Mabrouka Bessghaier, Shimaa Ibrahim, Md. Rafiul Biswas, and Wajdi Zaghouani. 2025. Marsadlab at arahealthqa: Hybrid contextual-lexical fusion with arabert for question and answer categorization. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1*.

Sajib Bhattacharjee, Ratnajit Dhar, Kawsar Ahmed, and Mohammed Moshiul Hoque. 2025. Binary_bunch at arahealthqa track 1: Advancing multi-label question and answer categorization with data augmentation and fine-tuned transformer model. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1*.

Jaya Chaturvedi, Sumithra Velupillai, Robert Stewart, and Angus Roberts. 2023. Identifying mentions of pain in mental health records text: a natural language processing approach. *arXiv preprint arXiv:2304.01240*.

Adiba Fairooz Chowdhury and MD Sagor Chowdhury. 2025. Quasar at arahealthqa track 1: Leveraging zero-shot large language models for question and answer categorization in arabic mental health. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1*.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E. Shamout. 2025. Medarabiq: Benchmarking large language models on arabic medical tasks. *Preprint*, arXiv:2505.03427.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Fatimah Emad Elden and Mumina Abukar. 2025. Sakinah-ai at mentalqa: A comparative study of few-shot, optimized, and ensemble methods for arabic mental health question classification. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1*.

Fatima Emad Eldin and Mumina Abukar. 2025. Medlingua at medarabiq2025: Zero- and few-shot prompting of large language models for arabic medical qa. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 2*.

Nejood A. Bin Eshaq. 2025. Mindllm at arahealthqa 2025 track 1: Leveraging large language models for mental health question answering. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1*.

Yanjun Gao, Dmitriy Dligach, Timothy Miller, John Caskey, Brihat Sharma, Matthew M. Churpek, and Majid Afshar. 2023. Dr.bench: Diagnostic reasoning benchmark for clinical natural language processing. *Journal of Biomedical Informatics*, 138:104286.

Muskan Garg, Manas Gaur, Raxit Goswami, and Sunghwan Sohn. 2023. Lost: A mental health dataset of low self-esteem in reddit posts. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3854–3859. IEEE.

Muskan Garg, Chandni Saxena, Sriparna Saha, Veena Krishnan, Ruchi Joshi, and Vijay Mago. 2022. Cams: An annotated corpus for causal analysis of mental health issues in social media posts. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6387–6396.

Gemini Team: Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, and Damien Vincent et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Soumitra Ghosh, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Cease, a corpus of emotion annotated suicide notes in english. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1618–1626.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, and Akhil Mathur et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Baraa Hikal. 2025. Medgapgab at arahealthqa: Modular llm assignment for gaps and gabs in arabic medical question answering. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 2*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Preprint*, arXiv:2009.13081.

Muhammad Khubayeeb Kabir, Maisha Islam, Anika Nahian Binte Kabir, Adiba Haque, and Md Khalilur Rhaman. 2022. Detection of depression severity using bengali social media posts on mental health: Study using natural language processing techniques. *JMIR Form. Res.*, 6(9):e36118.

AbdulRahman A. Morsy, Saad Mankarious, and Ayah Zirikly. 2025. Sindbad at arahealthqa track 1: Leveraging large language models for mental health q&a. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1*.

Ahmad Mustapha, Hadi Al-Khansa, Hadi Al-Mubasher, Aya Mourad, Ranam Hamoud, Hasan El-Husseini, Marwah Al-Sakkaf, and Mariette Awad. 2024. Arastem: A native arabic multiple choice question benchmark for evaluating llms knowledge in stem subjects. *Preprint*, arXiv:2501.00559.

Aryan Rastogi, Qian Liu, and Erik Cambria. 2022. Stress detection from social media articles: New dataset benchmark and analytical study. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Caroline Sabty, Mohamad Rasmy, Mohamed Eyad Badran, Nourhan Sakr, and Alia El Bolock. 2025. Fahmni at arahealthqa track 1: Multi-agent retrieval-augmented generation and multi-label classification for arabic mental health q&a. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1*.

Guangyao Shen, Jia Jia, Liqiang Nie, Fuli Feng, Cunjun Zhang, Tianrui Hu, Tat-Seng Chua, and Wenwu Zhu. 2017. Depression detection via harvesting social media: A multimodal dictionary learning solution. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3838–3844, California. International Joint Conferences on Artificial Intelligence Organization.

Hao Sun, Zhenru Lin, Chujie Zheng, Siyang Liu, and Minlie Huang. 2021. PsyQA: A chinese dataset for generating long counseling text for mental health support. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mohamed Tarek, Saif Ahmed, and Mohamed Basem. 2025. Msa at arahealthqa 2025 shared task: Enhancing llm performance for arabic clinical question answering through prompt engineering and ensemble learning. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 2*.

Elsbeth Turcan and Kathleen Mckeown. 2019. Dreaddit: A reddit dataset for stress analysis in social media. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 97–107.

Zhen Xu, Sergio Escalera, Adrien Pavão, Magali Richard, Wei-Wei Tu, Quanming Yao, Huan Zhao, and Isabelle Guyon. 2022. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns*, 3(7):100543.

Mohamed Zaytoon, Ahmed Salem, Ahmed Sakr, and Hossam Elkordi. 2025. Araminds at arahealthqa 2025: A retrieval-augmented generation system for fine-grained classification and answer generation of arabic mental health q&a. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), AraHealthQA Shared Task Track 1*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

NYUAD at AraHealthQA Shared Task: Benchmarking the Medical Understanding and Reasoning of Large Language Models in Arabic Healthcare Tasks

Nouar AlDahoul
Computer Science Department
New York University
Abu Dhabi, UAE
nouar.aldahoul@nyu.edu

Yasir Zaki
Computer Science Department
New York University
Abu Dhabi, UAE
yasir.zaki@nyu.edu

Abstract

Recent progress in large language models (LLMs) has showcased impressive proficiency in numerous Arabic natural language processing (NLP) applications. Nevertheless, their effectiveness in Arabic medical NLP domains has received limited investigation. This research examines the degree to which state-of-the-art LLMs demonstrate and articulate healthcare knowledge in Arabic, assessing their capabilities across a varied array of Arabic medical tasks. We benchmark several LLMs using a medical dataset proposed in the Arabic NLP AraHealthQA challenge in MedArabiQ2025 track. Various base LLMs were assessed on their ability to accurately provide correct answers from existing choices in multiple-choice questions (MCQs) and fill-in-the-blank scenarios. Additionally, we evaluated the capacity of LLMs in answering open-ended questions aligned with expert answers. Our results reveal significant variations in correct answer prediction accuracy and low variations in semantic alignment of generated answers, highlighting both the potential and limitations of current LLMs in Arabic clinical contexts. Our analysis shows that for MCQs task, the proposed majority voting solution, leveraging three base models (Gemini Flash 2.5, Gemini Pro 2.5, and GPT o3), outperforms others, achieving up to 77% accuracy and securing first place overall in the challenge[1] (Alhuzali et al., 2025). Moreover, for the open-ended questions task, several LLMs were able to demonstrate excellent performance in terms of semantic alignment and achieve a maximum BERTScore of 86.44%.

1 Introduction

Medicine relies heavily on complex reasoning, spanning tasks from diagnostic decision-making to treatment planning, especially when patient outcomes depend on understanding multi-factorial conditions (Qiu et al., 2024; Huang et al., 2025). Differential diagnosis involves generating and narrowing down possible diagnoses using clinical evidence, requiring both extensive medical knowledge and logical reasoning to evaluate multiple hypotheses.

LLMs have demonstrated superior performance across various domains and applications, such as article debiasing (Kuo et al., 2025), content moderation (AlDahoul et al., 2024b), and political leaning detection (AlDahoul et al., 2024a). In the healthcare domain, LLMs are reshaping the landscape of healthcare by transforming the way consultations, diagnoses, and treatment plans are delivered (Yang et al., 2023). They offer new avenues for improving patient education through dynamic, conversational interactions, thereby enhancing both accessibility and patient autonomy. Beyond direct patient care, LLMs also show promise in supporting medical training and streamlining administrative responsibilities, including the generation of clinical notes, referral letters, and discharge summaries (Yang et al., 2023).

Most existing benchmarks focus on English, leaving a gap in evaluating Arabic LLMs for healthcare due to the lack of high-quality clinical datasets, Arabic's linguistic diversity, and the limited performance of multilingual models in domain-specific tasks (Daoud et al., 2025). To fill these gaps, there is an increasing demand for frameworks that evaluate LLM performance in clinical tasks for Arabic-speaking communities. Our analyses and experiments center around the following research questions: **RQ1**: Do state-of-the-art proprietary base LLMs perform well in Arabic medical tasks? **RQ2**: To what extent do state-of-the-art proprietary base LLMs with reasoning capacity excel in Arabic medical tasks? **RQ3**: Do open-source-based Arabic LLMs perform well in Arabic medical tasks? and **RQ4**: How does majority voting among several LLMs enhance performance in Arabic medical

[1] https://www.codabench.org/competitions/8967/#/results-tab

Proceedings of The Third Arabic Natural Language Processing Conference, pages 119–125
November 8-9, 2025 ©2025 Association for Computational Linguistics

tasks?

We address **RQ1** by running the APIs of several LLMs, such as Claude Opus, Grok 3, Deepseek v3, Llama 4 Maverick, GPT-4o-mini, and GPT-4o. To answer **RQ2**, we utilized APIs of state-of-the-art LLMs with reasoning capabilities such as GPT-o3, Gemini Flash 2.5, and Gemini Pro 2.5. Moreover, to address **RQ3**, we ran Falcon 3, Fanar, and Allam. Additionally, to answer **RQ4**, we calculated the majority vote among the predictions of three LLMs.

2 Related Work

BioBERT (Lee et al., 2020), SCIBERT (Beltagy et al., 2019), and PubMedBERT (Gu et al., 2021) improved biomedical NLP by training on domain-specific corpora, thereby outperforming the general BERT model (Yang et al., 2023). Building on this, ClinicalBERT (Alsentzer et al., 2019) enhanced performance on medical tasks by fine-tuning BERT and BioBERT using the MIMIC-III clinical dataset. Expanding further, GatorTrona (Yang et al., 2022). significantly larger model trained from scratch on extensive clinical and biomedical text—demonstrated strong results across a wide range of clinical NLP tasks (Yang et al., 2023).

Various benchmarks have been developed to evaluate LLMs' proficiency in medical reasoning and knowledge (Huang et al., 2025; Zuo et al., 2025). However, significant challenges persist, ranging from ethical and safety concerns to the risk of biased outputs and inconsistent performance across different languages and cultural settings (Yang et al., 2023; Nazi and Peng, 2024; Daoud et al., 2025).

To advance medical LLMs, researchers have increasingly focused on creating multilingual medical datasets (Qiu et al., 2024). They introduced MMedC, a 25.5-billion-token multilingual medical corpus, and MMedBench, a multilingual QA benchmark with rationales. By fine-tuning Llama 3 (8B), they found it outperformed all other open-source models and approached GPT-4 performance. However, Arabic was not one of the languages included (Qiu et al., 2024).

Arabic medical benchmarks are limited and mostly focused on question-answering tasks. While resources like MMLU (Hendrycks et al., 2020), AraSTEM (Mustapha et al., 2024), and AraMed (Alasmari et al., 2024) offer valuable contributions, they do not fully cover the breadth of Arabic medical tasks, highlighting the need for more comprehensive benchmarking efforts. The previous issue was addressed by the MedArabiQ benchmark (Daoud et al., 2025).

3 Materials and Methods

3.1 Dataset Overview

The medical data used in this work is the main dataset utilized in the AraHealthQA shared task in the MedArabiQ2025 track (Alhuzali et al., 2025) under one of the Arabic NLP challenges. It focuses on modern standard Arabic (MSA) and consists of 700 diverse clinical samples, covering both structured medical knowledge assessments and real-world patient-doctor interactions (Daoud et al., 2025; Alhuzali et al., 2025). The dataset has multiple-choice and open-ended questions that are distributed as follows:

- a random set of 100 multiple-choice questions to evaluate the models' medical understanding.
- a set of 100 multiple-choice questions with bias injected to evaluate how LLMs handle ethical or culturally sensitive scenarios.
- a set of 100 fill-in-the-blank questions with choices to evaluate the model's ability to recognize correct answers, reducing the reliance on generative capabilities.
- a set of 100 fill-in-the-blank questions without choices to assess LLMs' reasoning and generation capabilities.
- a set of 100 patient-doctor Q&As selected from AraMed (Alasmari et al., 2024) to evaluate LLMs with online real-world scenarios from medical discussion forums.
- a 100 Q&As with grammatical error correction to handle inflectional patterns and prepare the dataset for grammatical correction.
- a 100 Q&As with LLM Modifications to mitigate potential model memorization and to assess the model's reasoning and adaptability.

The previous 700 examples were used for evaluation of LLMs. Later, another set of 200 examples (100 MCQs and 100 open-ended questions) was released for testing the LLMs' reasoning and understanding.

3.2 Methods

We have evaluated state-of-the-art base LLMs to identify the best in terms of correct answer match accuracy in MCQs task and alignment score of generated answers in open-ended questions task. This LLM can understand the questions, identify the correct answers utilizing its embedded knowledge and reasoning capability, and generate the answers that align with those of experts.

We started assessing several proprietary base LLMs for the MCQs task to evaluate the accuracy of the match between real and predicted answers. We used LLMs' APIs in the inference mode utilizing two different zero-shot prompts specialized for the MCQs task (Prompt 1 and Prompt 2) shown in the Appendix. The evaluated LLMs are: Gemini Flash 2.5, Gemini Pro 2.5[2] (Team et al., 2023), GPT-4o-mini[3], GPT-4o (Hurst et al., 2024), GPT o3[4], Grok 3[5], Claude 3 Opus[6], Deepseek v3 (Liu et al., 2024), and Llama 4 Maverick[7].

Later, we selected the two LLMs that have shown high performance in the MCQs task: Gemini Flash 2.5 and Gemini Pro 2.5 and utilized them in the open-ended question task. We also demonstrated the performance of small-sized LLMs such as GPT-4o-mini in this task. We utilized three different prompts specialized for open-ended tasks (Prompt 1, Prompt 2, and Prompt 3) which are also shown in the Appendix.

Additionally, open-source-based Arabic LLMs such as Falcon3 (Almazrouei et al., 2023) ("tiiuae/Falcon3-7B-Instruct")[8],[9], Fanar (Team et al., 2025) ("QCRI/Fanar-1-9B-Instruct")[10], and Allam (Bari et al., 2024)("ALLaM-AI/ALLaM-7B-Instruct-preview")[11] were assessed for both tasks.

We applied zero-shot prompting across all models and tasks, setting the temperature to 0 and top_p to 1 for all tasks to ensure deterministic responses. For the open-ended question task, BERTScore was used as an evaluation metric to measure alignment between generated and expert answers. For this purpose, we used the "XLM-RoBERTa-Large model" (Daoud et al., 2025), which was trained on multiple languages, including Arabic.

We also evaluated Arabic Falcon[12]. Since there is no API available for Arabic Falcon, we used the web interface to manually input questions into the chat version. We retained the history of previous questions to avoid clearing the context before each new query.

3.3 Results and Discussion

The results of the MCQs task using the proprietary LLMs are shown in Table 1. The dataset has MCQs related to understanding and reasoning. While understanding involves factual knowledge, reasoning mimics how doctors make decisions.

The medical reasoning capacity of GPT-o3, Gemini Flash 2.5, and Gemini Pro 2.5 makes them have superior performance compared to other LLMs. These simulate diagnostic thinking by combining multiple facts and using step-by-step reasoning to eliminate plausible but incorrect distractors in medical MCQs, which answers **RQ2**.

Model	Prompt	Accuracy%
GPT-4o-mini	1	49
GPT-4o	1	57
GPT-O3	1	72
Gemini Flash 2.5	1	73
Gemini Pro 2.5	1	75
GPT-O3	2	74
Gemini Flash 2.5	2	74
Gemini Pro 2.5	2	76
Majority voting	2	77
Grok 3	2	60
Claude 3 Opus	2	49
Falcon Arabic	2	38
Deepseek v3	2	56
Llama 4 Maverick	2	63

Table 1: Accuracy of different proprietary base LLMs using different prompts.

Even though Claude 3, Deepseek 3, Grok 3, and

[2] https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking

[3] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[4] https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf

[5] https://x.ai/news/grok-3

[6] https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf

[7] https://ai.meta.com/blog/llama-4-multimodal-intelligence/

[8] https://huggingface.co/blog/falcon3

[9] https://huggingface.co/tiiuae/Falcon3-7B-Instruct

[10] https://huggingface.co/QCRI/Fanar-1-9B-Instruct

[11] https://huggingface.co/ALLaM-AI/ALLaM-7B-Instruct-preview

[12] https://falcon-lm.github.io/blog/falcon-arabic/

Llama 4 Maverick possess strong reasoning capabilities, they exhibit modest performance on this task, likely due to limited medical knowledge or insufficient proficiency in Arabic, which addresses **RQ1** and **RQ2**. However, Llama 4 Maverick was the best among them in terms of accuracy (63%).

For sensitivity of prompt construction, we found that Prompt 2, which includes step-by-step or chain-of-thought reasoning, is generally better than simple Prompt 1 when it comes to answering medical MCQs.

The significant finding in this work is that current state-of-the-art proprietary LLMs exhibit limitations in their embedded medical knowledge of various Arabic medical tasks (maximum accuracy is 76% in Gemini Pro 2.5). The source of errors in the MCQ task may stem from misunderstanding of questions, lack of medical knowledge, or lack of medical reasoning capabilities.

To benefit from the capacity of each of three LLMs (GPT-O3, Gemini Flash 2.5, and Gemini Pro 2.5) in MCQs task, we applied a majority voting technique using the predictions from these LLMs, resulting in a final accuracy of 77%, which secured first place overall in the challenge, which answers **RQ4**.

The results of the open-ended questions task using proprietary LLMs are shown in Table 2. The dataset has questions labeled with answers. The LLMs should generate answers that are semantically aligned with reference answers.

Our finding indicates that reasoning LLMs such as Gemini Flash 2.5 and Gemini Pro 2.5 have structured answers that reduce hallucination and overconfidence, as the models are less likely to guess and more likely to justify their answers. As a result, their responses often align more closely with reference answers and perform better on semantic evaluation metrics like BERTScore, which answers **RQ2**. Furthermore, GPT-4o-mini shows good performance in terms of BERTScore.

Additionally, the three LLMs showed high sensitivity to prompts with variances in BERTScores. The maximum BERTScores were achieved by Prompt 3 that asked the LLMs to have modern standard Arabic in response, emphasized medically correct answers, and asked for concise answers that are not diluted with explanations, which usually tend to align more closely with reference answers.

Model	Prompt	BERTScore
Gemini Pro 2.5	1	0.8105
Gemini Flash 2.5	2	0.8364
GPT-4o-mini	2	0.8386
GPT-4o-mini	3	0.8581
Gemini Flash 2.5	3	0.8633
Gemini Pro 2.5	3	0.8644

Table 2: BERTScore of proprietary base LLMs using different prompts.

Table 3 shows the accuracy and BERTScore of several open-source base Arabic LLMs. Among the models, Allam demonstrates relatively better performance (39%) in MCQs task, while Falcon 3 gave the best BERTScore (0.8493). This experiment indicates a lack of medical knowledge and/or medical reasoning in the base open-source Arabic LLMs compared to proprietary ones, which addresses **RQ3**.

Model	Task	Accuracy %
Falcon 3	Task 1	36
Fanar	Task 1	31
Allam	Task 1	39
Model	**Task**	**BERTScore**
Falcon 3	Task 2	0.8493
Fanar	Task 2	0.8403
Allam	Task 2	0.8431

Table 3: Accuracy and BERTScore of different base Arabic LLMs.

Limitations

The first limitation is that multiple-choice and fill-in-the-blank with choice questions in the MedArabiQ2025 dataset are limited to only a few hundred examples. There is a clear need for larger, high-quality Arabic medical datasets to fine-tune LLMs and enhance their performance. Alternatively, storing extensive medical data in a vector database and employing retrieval-augmented generation (RAG) techniques could help retrieve more accurate and contextually relevant answers.

A second limitation of this work is the absence of bias detection and mitigation techniques during the preprocessing of questions before inputting them to LLMs. Incorporating such techniques could play a significant role in improving model performance and ensuring more reliable outputs.

The third limitation is that for open-ended and fill-in-the-blank questions without choices, we lack a robust metric for capturing semantic similarity.

In this work, we utilized BERTScore, which often yields similar values across different responses and fails to reflect subtle nuances in semantic alignment with the correct answers.

References

Ashwag Alasmari, Sarah Alhumoud, and Waad Alshammari. 2024. Aramed: Arabic medical question answering using pretrained transformer language models. 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 50–56.

Nouar AlDahoul, Talal Rahwan, and Yasir Zaki. 2024a. Polytc: a novel bert-based classifier to detect political leaning of youtube videos based on their titles. *Journal of Big Data*, 11(1):80.

Nouar AlDahoul, Myles Joshua Toledo Tan, Harishwar Reddy Kasireddy, and Yasir Zaki. 2024b. Advancing content moderation: Evaluating large language models for detecting sensitive content across text, images, and videos. *arXiv preprint arXiv:2411.17123*.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, et al. 2025. Arahealthqa 2025 shared task description paper. *arXiv preprint arXiv:2508.20047*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan Al-Rashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, et al. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E Shamout. 2025. Medarabiq: Benchmarking large language models on arabic medical tasks. *arXiv preprint arXiv:2505.03427*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Zhongzhen Huang, Gui Geng, Shengyi Hua, Zhen Huang, Haoyang Zou, Shaoting Zhang, Pengfei Liu, and Xiaofan Zhang. 2025. O1 replication journey–part 3: Inference-time scaling for medical reasoning. *arXiv preprint arXiv:2501.06458*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Chen Wei Kuo, Kevin Chu, Nouar AlDahoul, Hazem Ibrahim, Talal Rahwan, and Yasir Zaki. 2025. Neutralizing the narrative: Ai-powered debiasing of online news articles. *arXiv preprint arXiv:2504.03520*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Ahmad Mustapha, Hadi Al-Khansa, Hadi Al-Mubasher, Aya Mourad, Ranam Hamoud, Hasan El-Husseini, Marwah Al-Sakkaf, and Mariette Awad. 2024. Arastem: A native arabic multiple choice question benchmark for evaluating llms knowledge in stem subjects. *arXiv preprint arXiv:2501.00559*.

Zabir Al Nazi and Wei Peng. 2024. Large language models in healthcare and medical domain: A review. In *Informatics*, volume 11, page 57. MDPI.

Pengcheng Qiu, Chaoyi Wu, Xiaoman Zhang, Weixiong Lin, Haicheng Wang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024. Towards building multilingual language model for medicine. *Nature Communications*, 15(1):8384.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, et al. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

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*.

Rui Yang, Ting Fang Tan, Wei Lu, Arun James Thirunavukarasu, Daniel Shu Wei Ting, and Nan Liu. 2023. Large language models in health care: Development, applications, and challenges. *Health Care Science*, 2(4):255–263.

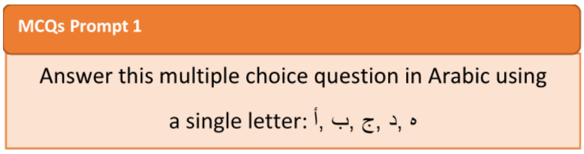
Xi Yang, A Chen, N PourNejatian, HC Shin, KE Smith, C Parisien, C Compas, C Martin, AB Costa, MG Flores, et al. 2022. A large language model for electronic health records. npj digital medicine, 5 (1), 1–9. *URL http://dx. doi. org/10.1038/s41746-022-00742-2*.

Yuxin Zuo, Shang Qu, Yifei Li, Zhangren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. 2025. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. *arXiv preprint arXiv:2501.18362*.

A Appendix

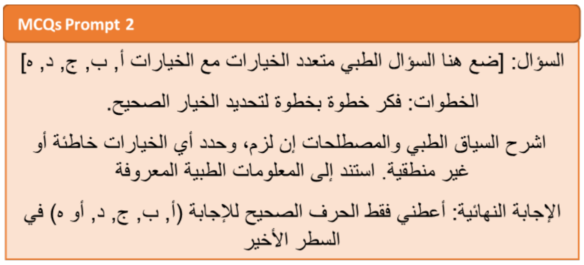
A.1 Prompts used

This work employed different prompts for the two tasks: the orange-colored prompts were used for Task 1 (MCQs), while the red, green, and blue prompts were used for Task 2 (open-ended questions).



MCQs Prompt 1

Answer this multiple choice question in Arabic using
a single letter: أ, ب, ج, د, ه



MCQs Prompt 2

السؤال: [ضع هنا السؤال الطبي متعدد الخيارات مع الخيارات أ, ب, ج, د, ه]

الخطوات: فكر خطوة بخطوة لتحديد الخيار الصحيح.

اشرح السياق الطبي والمصطلحات إن لزم، وحدد أي الخيارات خاطئة أو
غير منطقية. استند إلى المعلومات الطبية المعروفة

الإجابة النهائية: أعطني فقط الحرف الصحيح للإجابة (أ, ب, ج, د, أو ه) في
السطر الأخير

Open-ended questions' Prompt 1

You are a knowledgeable and concise medical expert. Provide a high-quality answer to the following open-ended medical question.
Your response should:
Begin with a direct, evidence-based answer.
Elaborate on the mechanisms, relevant anatomy or physiology, and clinical significance.
Use clear, professional medical language.
Question:
[Insert your medical question here]

Open-ended questions' Prompt 2

You are a knowledgeable and concise medical expert. Provide a high-quality answer to the following open-ended medical question.

Open-ended questions’ Prompt 3

You are a knowledgeable and concise medical expert.
Your task is to generate a concise, accurate, and medically correct answer in Modern Standard Arabic.
Do not include explanations—just provide the best possible answer based on your knowledge.

MedLingua at MedArabiQ2025: Zero- and Few-Shot Prompting of Large Language Models for Arabic Medical QA

Fatimah Emad Eldin, Mumina Abukar
Cairo University, The University of South Wales
{12422024441586@pg.cu.edu.eg, 74108361@students.southwales.ac.uk}

Abstract

This paper details the system developed by team MedLingua for the MedArabiQ2025 Shared Task, specifically participating in Track 2, Sub-Task 1: Multiple Choice Question Answering. Our approach centered on evaluating the zero-shot and few-shot capabilities of various Large Language Models (LLMs) on Arabic medical questions, as fine-tuning was not permitted. We systematically tested a range of models, from general-purpose state-of-the-art LLMs like Google's Gemini 2.5 Pro to specialized medical models such as BiMediX2 and MedGemma. Our findings reveal that advanced, general-domain models significantly outperform specialized medical LLMs that are not optimized for Arabic. Our best performing system, using Gemini 2.5 Pro, achieved an accuracy of 78% in the development set and 74% on the blind test set, securing the 3rd place on the official competition leaderboard.

1 Introduction

The MedArabiQ2025 shared task addresses the critical need for robust natural language understanding systems in the Arabic medical domain (Abu Daoud et al., 2025). Our team, MedLingua, participated in Track 2, Sub-Task 1, which focuses on Multiple Choice Question Answering (MCQA). This task is vital for developing clinical decision support systems and educational tools tailored to Arabic-speaking healthcare professionals and students. The primary challenge lies in the complexity of medical language and the relative scarcity of high-quality Arabic medical datasets and models compared to English. Given the constraint that participants could not fine-tune models on the provided data, our core strategy was to leverage the in-context learning abilities of existing LLMs. We employed both zero-shot and few-shot prompting techniques to guide various models toward the correct answer.

Our key finding was the pronounced performance gap between large, multilingual general-purpose models and the available specialized medical LLMs. The former demonstrated superior understanding of the Arabic questions, while many of the latter struggled with the language or failed to adhere to the task's constraints. Our best system achieved 74% accuracy on the blind test set, demonstrating the effectiveness of modern LLMs in this zero-resource fine-tuning scenario. To ensure reproducibility and facilitate future research in Arabic medical question answering, we make all experimental code publicly available on GitHub [1].

2 Background and Related Work

Question answering in the medical domain is a well-established research area (Pampari et al., 2018). However, most work, including the development of specialized models like Palmyra-Med Writer Engineering team (2024) and Med-PaLM (Singhal et al., 2023), has been overwhelmingly focused on English. While models like BiMediX2 (Mullappilly et al., 2024) have emerged to address the bilingual (Arabic-English) need, the field is still nascent. The MedArabiQ benchmark (Abu Daoud et al., 2025) is a crucial step in spurring research in this area. Our work contributes by providing a comprehensive evaluation of how current SOTA generalist and specialist LLMs perform on this new Arabic benchmark without task-specific fine-tuning.

3 Data

3.1 Shared Task Data

The MedArabiQ2025 MCQA sub-task is framed as a classification problem where the system receives a question in Arabic and must return the single

[1] https://github.com/astral-fate/AraHealthQA-2025-MedArabiQA

Proceedings of The Third Arabic Natural Language Processing Conference, pages 126–139
November 8-9, 2025 ©2025 Association for Computational Linguistics

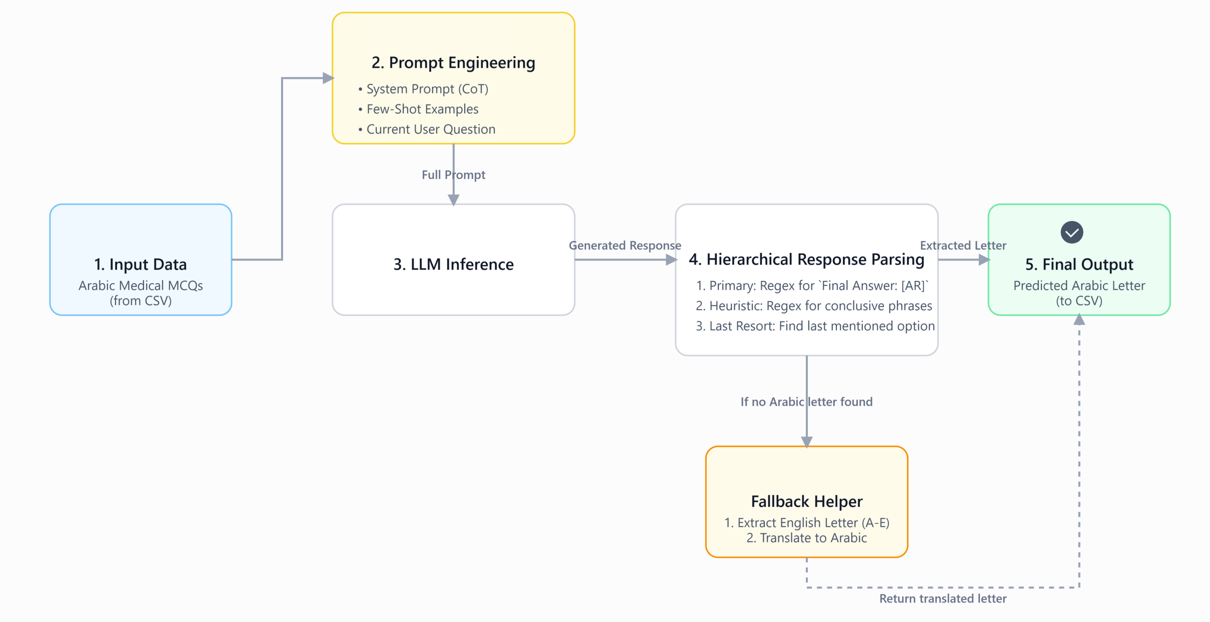


Figure 1: Overview of the system architecture for Arabic Medical QA.

letter corresponding to the correct answer from a list of choices (Alhuzali et al., 2025).

The organizers provided three distinct datasets for model development and validation, each containing 100 questions. The questions were sourced from medical exams and categorized into 12 medical specialties.

3.2 Validation Dataset

The validation data was split into three types, which we used for iterative testing and model selection:

- **Multiple Choice Questions (MCQ):** A standard set of multiple-choice questions.
- **Multiple Choice Questions with Bias (MCQ w/ Bias):** Questions designed with misleading phrasing to test model robustness.
- **Fill-in-the-Blank (FITB) with Choices:** Questions presented in a fill-in-the-blank format.

3.3 Test Dataset

The final evaluation was performed on a blind test set containing 100 questions. This dataset was a combination of all three question types from the validation set and was used to determine the final competition rankings.

4 Methodology

Our approach for Arabic medical question answering (QA) leverages in-context learning through various Large Language Models (LLMs), given the constraint against fine-tuning. The system architecture, designed to process Arabic medical multiple-choice questions (MCQs), is detailed in Figure 1.

4.1 Prompt Engineering and System Architecture

Our methodology centered on carefully structured prompt engineering to guide LLMs in a zero-shot or few-shot setting. The architecture can be broken down into five key stages:

1. **Input Data**: The process begins with loading Arabic medical MCQs from a CSV file.
2. **Prompt Engineering**: A full prompt is dynamically constructed by combining a system prompt, few-shot examples (if applicable), and the current question.
3. **LLM Inference**: The prompt is sent to an LLM for processing.
4. **Hierarchical Response Parsing**: The model's response is parsed using a multi-step process to extract the final answer.
5. **Final Output**: The extracted Arabic letter is saved to an output CSV for evaluation.

4.2 Chain-of-Thought (CoT) and Few-Shot Prompting

A key component of our strategy was the use of Chain-of-Thought (CoT) prompting.

Prompt Type	Prompt Structure
Few-Shot(e.g., MedGemma, Qwen)	`SYSTEM_PROMPT + FEW_SHOT_EXAMPLES + USER_QUESTION`
Zero-Shot (e.g., BioMistral)	`SYSTEM_PROMPT + USER_QUESTION`

Table 1: Comparison of prompt structures for few-shot and zero-shot learning.

We instructed models to first perform a step-by-step reasoning process within a <thinking> block before providing the final answer. An example of the Chain-of-Thought (CoT) prompt structure used for few-shot learning is provided in Appendix B (Table 4).

4.3 Zero-Shot vs. Few-Shot Strategies

Our approach involved testing both few-shot and zero-shot prompting strategies to determine the most effective method for each model. The fundamental difference in these approaches lies in the inclusion of examples within the prompt, as illustrated in Table 1.

4.3.1 The Case of BioMistral: When Few-Shots Fail

A notable example was **BioMistral** (Labrak et al., 2024). When provided with few-shot examples in Arabic, its output became nonsensical, generating repetitive, meaningless Arabic words. However, when we switched to a **zero-shot** approach (removing the examples), its behavior changed dramatically. Although it did not produce reasoning in Arabic, it performed the reasoning correctly in English and concluded with the correct Final Answer: format. This highlights that for some models, few-shot examples can confuse rather than guide.

4.4 Model Selection and Implementation

We experimented with two main categories of models:

1. **General-Purpose LLMs:** Models like Google's Gemini 2.5 Pro, Mixtral, Llama 3, and Qwen (Qwen Team, 2025), accessed via APIs (DeepMind AI Studio [2], NVIDIA NIM inference microservices API[3], Groq [4].)
2. **Specialized Medical LLMs:** Models like BiMediX2 (Mullappilly et al., 2024), MedGemma (Sellergren et al., 2025), BioMistral (Labrak et al., 2024), OpenBioLLM (Pal and Sankarasubbu, 2024), and Palmyra-Med Writer Engineering team (2024),

General-purpose LLMs (Gemini, Qwen, etc.) were accessed via APIs from DeepMind, NVIDIA, and Groq. For specialized models, MedGemma, BioMistral, and OpenBioLLM were accessed via Hugging Face; Palmyra-Med via the NVIDIA NIM API; and BiMediX2 was run locally on a Google Colab Pro+ A100 GPU.

5 Results

Our experiments revealed a striking performance gap, with large, general-purpose LLMs consistently outperforming specialized medical models on Arabic medical question answering. Our final submission, using Gemini 2.5 Pro, achieved **74% accuracy on the blind test set**, securing the 3rd place on the official competition leaderboard. Table 3 shows the performance of all 11 models we evaluated on the final blind test set.

For a more granular error analysis of the 100-question blind test set, the manual categorization of each question into 12 medical specialties was performed by co-author Dr. Mumina Abukar, MD, MScPH. This allowed us to precisely identify model weaknesses. Analysis of the categorized test set revealed that certain medical domains were universally more difficult for the models. The detailed error distribution by medical category and the accuracy versus execution time analysis are presented in Appendix A (see Figures 2a and 2b).

The primary sources of errors remained consistent with our development set findings: incorrect medical reasoning and output formatting failures.

5.1 Error Distribution on the Test Set

Table 2 details the error counts for the five highest-scoring models across the five most challenging medical categories, identified by the highest total number of errors across all tested models. Physiology emerged as the most difficult category, where even top models struggled. Notably, Gemini 2.5 Pro demonstrated the most robust performance, registering the lowest error count in three of the five

[2] https://aistudio.google.com/

[3] https://build.nvidia.com/models/

[4] https://console.groq.com/

Category	Gemini	MedGemma	Colosseum	Palmyra-Med	Llama3 70B
Physiology	6	13	12	11	14
Ophthalmology	4	7	5	9	9
Oncology	4	6	7	6	5
Biochemistry	4	4	5	5	6
Neurosurgery	1	4	6	6	7

Table 2: Focused Error Analysis: Error counts for the top 5 performing models in the 5 most error-prone medical categories on the blind test set.

Model	Test Accuracy
Gemini 2.5 Pro	**74%**
Qwen	67%
MedGemma	53%
Colosseum	51%
Palmyra-Med	49%
Llama3 70B	45%
BiMediX2	37%
Mixtral	21%
OpenBioLLM	21%
Biomistral	19%
DeepSeek	17%

Table 3: Performance of all evaluated models on the Blind Test set. Our final submission used Gemini 2.5 Pro.

most challenging categories: Neurosurgery (1 error), and tying for the lowest in Oncology (4 errors) and Biochemistry (4 errors). This highlights its strong reasoning capabilities even in complex domains.

6 Discussion

The pronounced performance gap between large, generalist LLMs and their specialized medical counterparts on the blind test set is the key finding of this work. The superior performance of models like Gemini 2.5 Pro (74%) and Qwen (67%), can be attributed to their advanced multilingual capabilities and vast general knowledge. These features appear to compensate for the lack of specific medical fine-tuning, especially when handling nuanced Arabic medical questions.

Our detailed error analysis of the test set reinforces this conclusion. The annotation of the test set questions into 12 medical specialties was manually performed by co-author Dr. Mumina Abukar, MD, MScPH, leveraging her expertise in the medical field. During this process, it became apparent that some questions, particularly those related to study design and data collection, did not fit precisely within the original 12 medical categories in Appendix I (Table 13) shows examples of such questions, which were categorized as "Physiology" in the original dataset but are better described as "Research Methodology". This potential mismatch could impact the fine-grained error analysis; however, for consistency with the original dataset structure, we adhered to the provided 12 categories for our evaluation.

The specialized models were largely hindered by a "language barrier." For instance, MedGemma's relatively high error rate in Physiology (13 errors, as shown in Table 2) suggests its specialized training did not effectively transfer to the Arabic context. This necessitated a translation-based approach for English-centric models like Palmyra-Med, which introduces potential information loss and likely limited their performance. BiMediX2, the only dedicated bilingual model tested, showed promise but was not competitive with the scale and reasoning power of top-tier generalist models on this task.

This outcome underscores a critical consideration for applying LLMs in specialized, non-English domains: strong foundational language understanding is a prerequisite for effective domain-specific reasoning. The test set results clearly show that Gemini's robust grasp of Arabic allowed it to apply its reasoning capabilities more effectively than models that were technically more specialized in medicine but weaker in the target language.

7 Conclusion

This work evaluated zero-shot and few-shot prompting strategies for Arabic medical question answering using general-purpose and specialized medical large language models. Our best-performing system achieved 74% accuracy on the MedArabiQ2025 blind test set using Gemini 2.5 Pro, securing the 3rd place on the official competition leaderboard.

Results demonstrate that advanced general-purpose models significantly outperformed specialized medical LLMs due to superior multilingual capabilities compensating for lack of domain-specific training.

Key limitations include language barriers hindering specialized models and potential dataset categorization inconsistencies. Future research should prioritize developing medical LLMs specifically trained on high-quality, large-scale Arabic medical corpora to bridge the identified performance gap between general and specialized models.

Acknowledgments

We thank the organizers of the MedArabiQ2025 shared task at New York University Abu Dhabi for creating this valuable benchmark and facilitating research in Arabic medical NLP.

References

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E. Shamout. 2025. MedArabiQ: Benchmarking Large Language Models on Arabic Medical Tasks. *arXiv preprint arXiv:2505.03427*.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, et al. 2025. AraHealthQA 2025 Shared Task Description Paper. *arXiv preprint arXiv:2508.20047*.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. BioMistral: A Collection of Open-Source Pretrained Large Language Models for Medical Domains. *arXiv preprint arXiv:2402.10373*.

Sahal Shaji Mullappilly, Mohammed Irfan Kurpath, Sara Pieri, Saeed Yahya Alseiari, Shanavas Cholakkal, Khaled Aldahmani, Fahad Khan, Rao Anwer, Salman Khan, Timothy Baldwin, and Hisham Cholakkal. 2024. BiMediX2: Bio-Medical EXpert LMM for Diverse Medical Modalities. *arXiv preprint arXiv:2412.07769*.

Ankit Pal and Malaikannan Sankarasubbu. 2024. OpenBioLLMs: Advancing Open-Source Large Language Models for Healthcare and Life Sciences. Hugging Face repository. `https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B`.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrQA: A Large Corpus for Question Answering on Electronic Medical Records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Qwen Team. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.

Andrew Sellergren et al. 2025. MedGemma Technical Report. *arXiv preprint arXiv:2507.05201*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972), 172-180.

Writer Engineering team. June 2024. Palmyra-Med-70b: A powerful LLM designed for healthcare. *Writer Engineering Blog*.

A Test Set Analysis: Error Distribution and Performance

B Example Prompt Structure

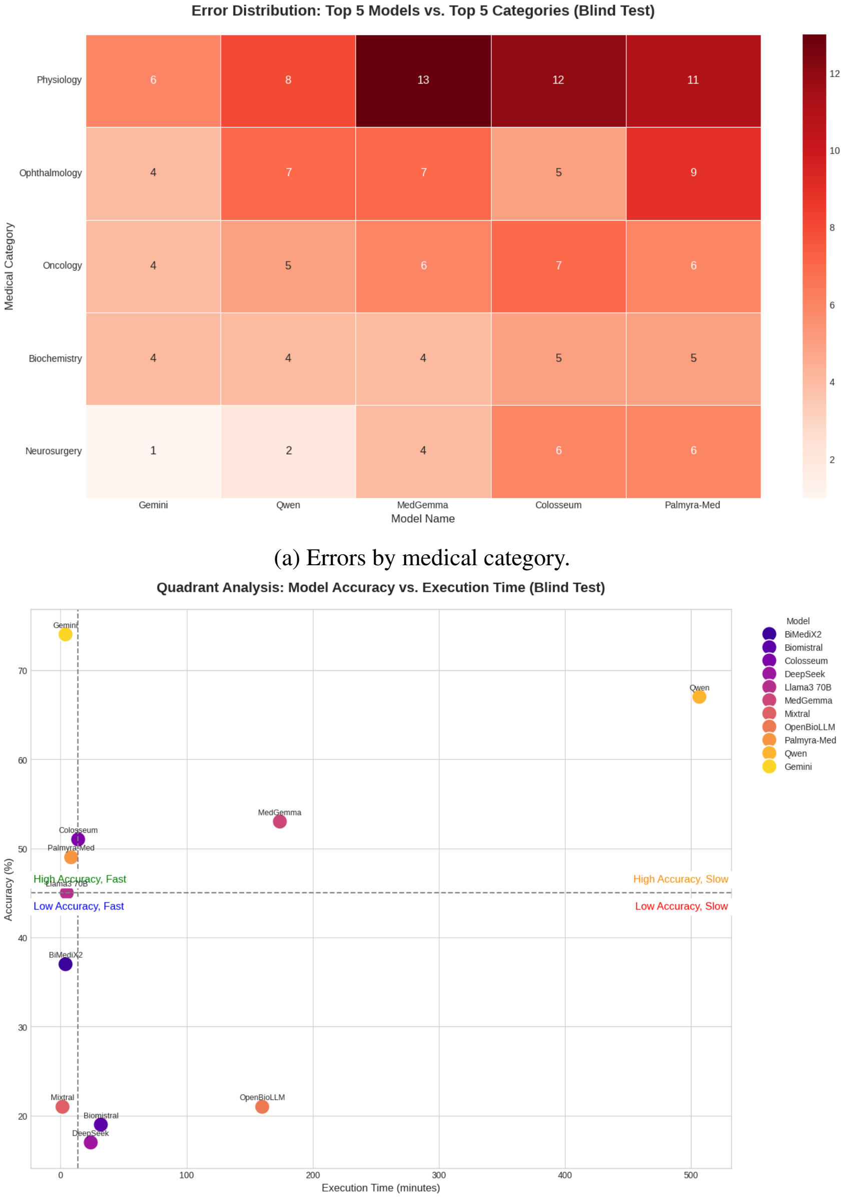
Table 4 illustrates the detailed Chain-of-Thought (CoT) prompt structure that was a key component of our methodology for the few-shot experiments, as referenced in Section B.

C Full Error Distribution on the Blind Test Set

Table 5 provides a comprehensive breakdown of the errors made by each of the 11 models evaluated on the blind test set. The questions were manually classified into 12 distinct medical specialties to facilitate this granular analysis.

D Summary of Model Performance on Development Datasets

This appendix provides a consolidated view of the performance of all evaluated models across the three distinct development datasets: Fill-in-the-Blank (FITB), standard Multiple Choice Question (MCQ), and MCQ with Bias. Table 6 summarizes the development accuracy for each model, highlighting the variance in performance depending on the question format and the presence of intentionally misleading phrasing.



(a) Errors by medical category.

(b) Accuracy vs. Execution Time.

Figure 2: A comparison of error distribution and performance on the blind test set.

E Analysis of the Fill-in-the-Blank (FITB) Task

This appendix presents a detailed Exploratory Data Analysis (EDA) of model performance on the "Fill-in-the-Blank with Choices" dataset. We analyze the overall accuracy, error distribution across medical specialties, and the relationship between model performance and inference time.

E.1 Model Performance Overview

The experiments revealed a wide range of performance. A clear hierarchy emerged, with a distinct group of high-performing models separating from the rest. **Gemini** achieved the highest accuracy at 84.0%, establishing itself as the top performer on this task. It was followed by a competitive tier including **MedGemma** (81.0%), **DeepSeek 70B** (78.0%), and **Colosseum** (75.0%). Conversely, several specialized models like **BioMistral** (15.0%) and **OpenBioLLM** (34.0%) struggled significantly. Table 7 summarizes the final accuracy and execution times for each model.

E.2 Error Analysis by Medical Category

To understand model weaknesses, we analyzed the distribution of errors across medical categories. The results show that certain domains were universally more difficult. The five categories with

Component	Example Content
System Prompt	You are an expert medical professional... Your task is to solve a multiple-choice question in Arabic. First, you will engage in a step-by-step thinking process in a `<thinking>` block... Second, after your reasoning, you MUST provide the final answer on a new line in the format: `Final Answer: [The single Arabic letter]`
User Question Example	املأ الفراغات... في حالة الانصباب الجنبي...

Ideal Assistant Response (with CoT)	`<thinking>`
	١. تحليل السؤال: يسأل السؤال عن دلالة انخفاض أو غياب الرجفان اللمسيْ... ٢. تقييم الخيارات: أ. تراكم السوائل؛ عزل الصوت... ٣. الاستنتاج: الخيار الأكثر دقة هو أن تراكم السوائل هو ما يسبب عزل الصوت...
	`</thinking>`
	Final Answer:
	أ

Table 4: Illustration of the Chain-of-Thought (CoT) prompt structure used in our few-shot experiments.

the highest total error counts were **OBGYN**, **Pulmonology**, **Cardiovascular System**, **Gastroenterology**, and **Neurology**. This suggests the questions in these fields may contain more complex terminology or require more nuanced clinical reasoning. Table 8 details the error counts for the top-performing models in these challenging categories.

E.3 Accuracy vs. Execution Time Analysis

The relationship between inference time and accuracy provides critical insights into model efficiency, as illustrated in the quadrant analysis in Figure 3c. We observe distinct performance archetypes:

1. **High Accuracy, Fast**: **Gemini** is the clear standout, occupying the top-left quadrant with the highest accuracy (84%) and a fast execution time. **DeepSeek 70B** (78%), **Colosseum** (75%) and **Palmyra-Med** (66%) also demonstrate strong efficiency.

2. **High Accuracy, Slow**: **MedGemma** resides in this category, achieving a high accuracy of 81% but requiring the longest execution time.

3. **Low Accuracy, Slow**: **BioMistral** is a notable example here, combining the lowest accuracy (15%) with a long execution time.

This analysis indicates that while more processing time can be beneficial, model architecture and optimization are paramount for achieving both speed and accuracy.

F Analysis of the Multiple Choice w/ Bias Task

This appendix presents a detailed Exploratory Data Analysis (EDA) of model performance on the "Multiple Choice with Bias" dataset. The objective is to identify which models were most resilient to the introduced bias and to pinpoint the medical categories where models struggled the most.

F.1 Model Performance Overview

The introduction of biased phrasing created a clear performance hierarchy among the models. Gemini 2.5 Pro demonstrated exceptional resilience to bias, achieving a top score of 75.0% and clearly separating itself from the other models. It was followed by Qwen (CoT), which also performed robustly with an accuracy of 68.0%. A competitive middle

Category	**Gemini**	**Qwen**	**MedGemma**	**Colosseum**	**Palmyra-Med**	**Llama3 70B**	**BiMediX2**	**Mixtral**	**OpenBioLLM**	**Biomistral**	**DeepSeek**
Biochemistry	4	4	4	5	5	6	9	10	9	8	10
Embryology	0	1	1	2	2	2	2	2	1	2	1
Histology	2	5	3	4	5	4	5	3	5	2	5
Microbiology	2	2	2	3	3	3	4	4	4	2	2
Neurosurgery	1	2	4	6	6	7	6	9	10	9	10
OBGYN	1	1	1	1	1	1	1	3	2	1	1
Oncology	4	5	6	7	6	5	5	11	9	10	9
Ophthalmology	4	7	7	5	9	9	11	10	11	10	12
Pediatrics	1	1	1	1	0	1	1	1	1	1	1
Pharmacology	0	0	0	2	2	2	4	3	4	4	4
Physiology	6	8	13	12	11	14	14	22	21	20	22
Pulmonology	1	1	1	1	1	1	1	1	2	2	1

Table 5: Full error distribution for all models across 12 medical categories on the 100-question blind test set.

Model’s Name	Fill in the Blank (Dev Acc)	Multiple Choice Question (Dev Acc)	Multiple Choice w/ Bias (Dev Acc)
Gemini 2.5 Pro	84%	78%	75%
qwen/qwen2-32b	83%	70%	68%
google/medgemma-27b-it	81%	55%	53%
deepseek-r1-distill-llama-70b (CoT)	78%	62%	53%
colosseum_355b_instruct_16k	75%	50%	45%
llama-3.3-70b-versatile	66%	57%	40%
palmyra-med-70b / 32k	66%	55%	35%
BiMediX2	52%	25%	31%
mixtral-8x22b-instruct-v2	40%	30%	19%
OpenBioLLM	34%	24%	18%
BioMistral	15%	19%	23%

Table 6: Comprehensive development accuracy results across the three development datasets.

Model	Accuracy (%)	Total Errors	Time (mins)
Gemini 2.5 Pro	84.00	16	50.00
MedGemma	81.00	19	176.07
DeepSeek 70B	78.00	22	25.00
Colosseum	75.00	25	14.72
Llama3 70B	69.00	31	27.20
Llama3 70B	66.00	34	27.33
Palmyra-Med	66.00	34	13.95
BiMediX2	52.00	48	10.07
Mixtral	40.00	60	15.90
OpenBioLLM	34.00	66	10.78
BioMistral	15.00	85	47.77

Table 7: Final performance summary for the Fill-in-the-Blank task.

model_name	Gemini 2.5 Pro	MedGemma	DeepSeek 70B	Colosseum	Llama3 70B
Category					
OBGYN	2	4	4	5	10
Pulmonology	4	5	5	5	9
Cardiovascular System	3	1	4	2	9
Gastroenterology	2	1	2	4	8
Neurology	1	3	1	2	7

Table 8: Error counts for top models in the five most challenging categories on the FITB task.

tier emerged, led by DeepSeek 70B (Groq) and MedGemma (Local), which tied at 53.0%.

F.2 Error Analysis by Medical Category

The five categories with the highest total error counts were **Embryology**, **Histology**, **Physiology**, **Biochemistry**, and **Microbiology**. This suggests that questions in these foundational science fields may be harder to answer correctly when potentially misleading information is present. The heatmap in Figure 4b shows that Gemini 2.5 Pro had the fewest errors in four of these five most difficult categories.

F.3 Accuracy vs. Execution Time Analysis

The quadrant analysis in Figure 4c highlights significant differences in efficiency. Gemini 2.5 Pro is the clear standout, occupying the "High Accuracy, Fast" quadrant and demonstrating the best balance of speed and performance. Qwen (CoT) falls into the "High Accuracy, Slow" category, delivering strong results but at a significant time cost. The remaining models form a cluster of lower-accuracy options, with DeepSeek 70B (Groq) offering the best performance among the faster, less accurate models.

Model	Accuracy (%)	Total Errors	Time (mins)
Gemini 2.5 Pro	75.00	25	4.00
Qwen (CoT)	68.00	32	81.68
DeepSeek 70B (Groq)	53.00	47	15.82
MedGemma (Local)	53.00	47	180.00
Colosseum	45.00	55	13.60
Llama3 70B (CoT)	40.00	60	20.25
Palmyra-Med	35.00	65	10.77
BiMediX2 (vLLM)	31.00	69	0.53
BioMistral (Fallback)	23.00	77	41.75
Mixtral	19.00	81	14.47
OpenBioLLM 8B (Local)	18.00	82	10.37

Table 9: Final performance summary for the MCQ with Bias task, based on the updated data.

model_name	Gemini 2.5 Pro	Qwen (CoT)	DeepSeek 70B (Groq)	MedGemma (Local)	Colosseum
Category					
Embryology	2	5	8	7	8
Histology	1	5	8	6	9
Physiology	3	6	7	9	9
Biochemistry	3	3	4	6	7
Microbiology	1	2	3	3	1

Table 10: Updated error counts for the new top 5 models in the five most challenging categories on the biased dataset.

G Analysis of the Multiple Choice Question (MCQ) Task

This appendix provides a detailed EDA of model performance on the standard "Multiple Choice Question" dataset. We examine the overall accuracy, error distribution, and the trade-offs between accuracy and processing time.

G.1 Model Performance Overview

The standard MCQ task revealed a clear performance hierarchy. Gemini 2.5 Pro established itself as the top-performing model with an impressive accuracy of 78%. It was followed by a tier of other strong models including Qwen (70%), DeepSeek (62%), Llama3 70B (57%), and both Palmyra-Med (55%) and MedGemma (55%). In contrast, some specialized models like Biomistral (19%) and OpenBioLLM (24%) struggled significantly.

G.2 Error Analysis by Medical Category

Some medical specialties were consistently more challenging for all models. The five categories accumulating the most errors were **Physiology**, **Histology**, **Embryology**, **Biochemistry**, and **Microbiology**. This indicates that questions in these foundational medical sciences likely require more complex reasoning or contain more specialized terminology. The error distribution for the top-performing models in these categories is detailed in Table 12.

G.3 Accuracy vs. Execution Time Analysis

The quadrant analysis of accuracy versus execution time in Figure 5c reveals four distinct performance profiles:

1. **High Accuracy / Fast**: This quadrant is led by the top performer, **Gemini**. Other strong models like **Qwen**, **DeepSeek**, **Llama3 70B**, and **Palmyra-Med** also fit here, offering high accuracy with efficient processing times.
2. **High Accuracy / Slow**: **MedGemma** stands alone in this category, achieving a respectable accuracy of 55% but requiring significantly more computational time (over 160 minutes).
3. **Low Accuracy / Fast**: Models like **Mixtral**, **BiMediX2**, and **OpenBioLLM** delivered results quickly but with lower accuracy scores.
4. **Low Accuracy / Slow**: **Biomistral** was the least efficient, combining low accuracy with a relatively slow execution time.

H Challenges in Manual Test Set Annotation

As mentioned in the Discussion, the manual categorization of the blind test set revealed that some questions did not align well with the provided 12 medical specialty categories. Table 13 lists five questions originally classified as "Physiology" that co-author Dr. Mumina Abukar, MD, MScPH, leveraging her expertise in the medical field, identified as belonging to "Research Methodology." This highlights a potential area for refinement in future iterations of the benchmark to ensure that the categories accurately reflect the question content, thereby improving the validity of category-based error analyses.

Model	Accuracy (%)	Total Errors	Time (mins)
Gemini 2.5 Pro	78.00	22	5.00
Qwen	70.00	30	40.00
DeepSeek	62.00	38	24.00
Llama3 70B	57.00	43	18.00
MedGemma	55.00	45	165.00
Palmyra-Med	55.00	45	8.00
Colosseum	50.00	50	24.00
Mixtral	30.00	70	2.00
BiMediX2	25.00	75	4.00
OpenBioLLM	24.00	76	12.00
Biomistral	19.00	81	33.00

Table 11: Final performance summary for the MCQ task. Total errors are based on a dataset size of 100 questions.

Category	Gemini	Qwen	DeepSeek	Llama3 70B	Palmyra-Med
Physiology	4	5	7	8	7
Histology	1	4	3	8	5
Embryology	1	2	8	8	7
Biochemistry	3	3	3	5	6
Microbiology	1	4	3	2	6

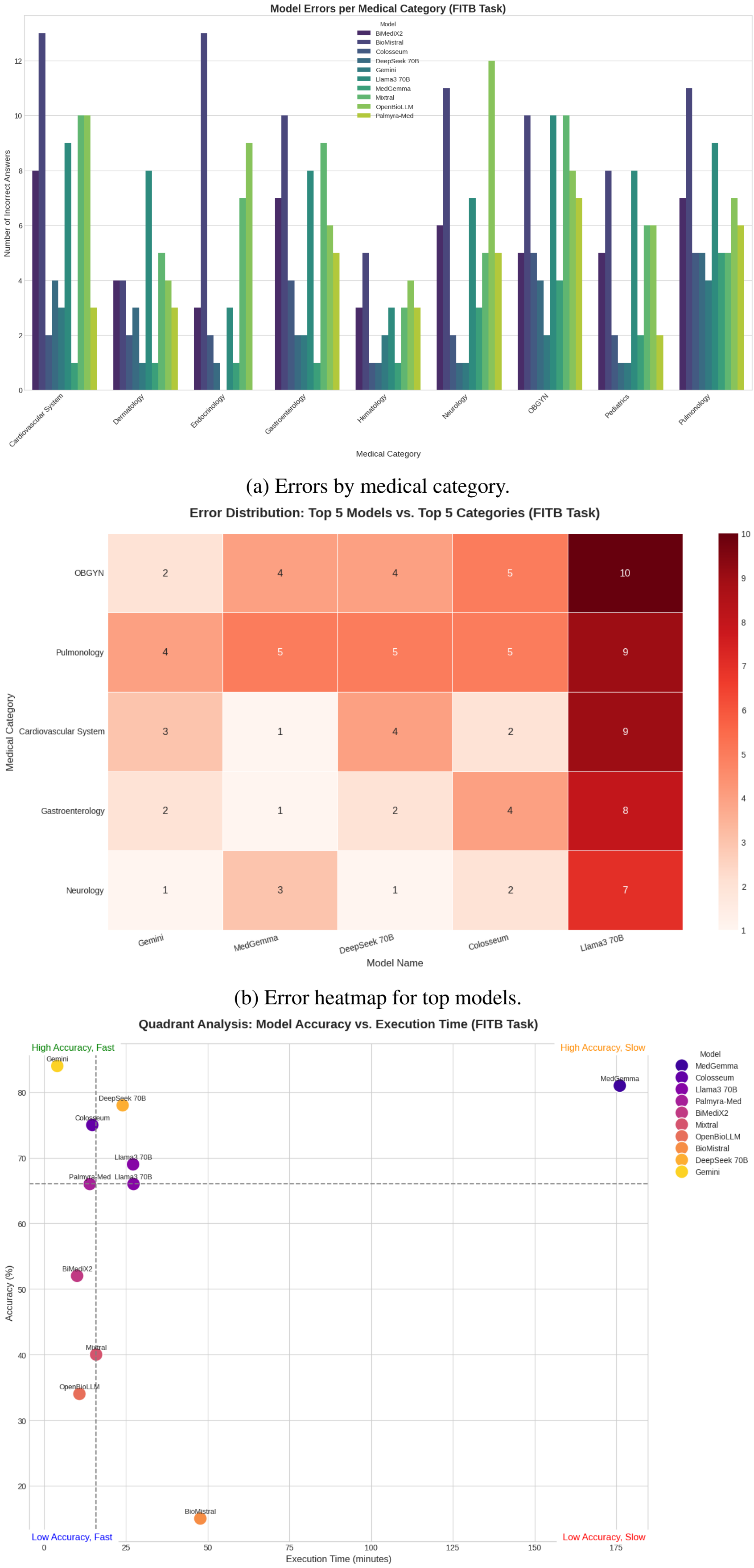
Table 12: Error counts for the top 5 models in the five most challenging categories on the MCQ task.

I Challenges in Manual Test Set Annotation

As mentioned in the Discussion, the manual categorization of the blind test set revealed that some questions did not align well with the provided 12 medical specialty categories. Table 13 lists five questions originally classified as "Physiology" that co-author Dr. Mumina Abukar, MD, MScPH, leveraging her expertise in the medical field, identified as belonging to "Research Methodology." This highlights a potential area for refinement in future iterations of the benchmark to ensure that the categories accurately reflect the question content, thereby improving the validity of category-based error analyses.

ID	Question	Original Category	Expert's Proposed Category
34	فيما يتعلق بالعينة العشوائية البسيطة، يُعد ...... ليس من خصائصها. أ. أبسط أنواع العينات ب. يتم اختيار الأفراد بإجراء القرعة ج. قيام طبيب بإجراء دراسة على مرضى مراجعين له مثال عليها د. تستخدم في حالة تجانس المجتمع	Physiology	Research Methodology
49	أساليب جمع البيانات تتضمن: أ. الاتصالات الهاتفية ب. المواقع الاجتماعية ج. استخدام الإنترنت د. كل ما سبق صحيح	Physiology	Research Methodology
54	من أساليب العينة العشوائية البسيطة . أ. اجراء قرعة عندما يكون حجم العينة صغيرا ب. استخدام جداول الأرقام العشوائية ج. عندما يكون حجم العينة كبيرة د. ا+ـ	Physiology	Research Methodology
77	من مزايا طريقة المواجهة أو المقابلة الشخصية ما عدا: أ. الحصول على إجابات دقيقة ومراقبة ردود ب. ارتفاع نسبة المستجيبين ج. كلفتها عالية د. تستخدم في المجتمعات التي ترتفع فيها نسبة الأمية	Physiology	Research Methodology
85	ال ...... هي من أقدم الطرق المستخدمة لجمع البيانات لمراقبة الكثير من الظواهر لا سيما التي يصعب السؤال عنها:	Physiology	Research Methodology

Table 13: Examples of questions from the blind test set with proposed category corrections. These questions were originally categorized under Physiology.

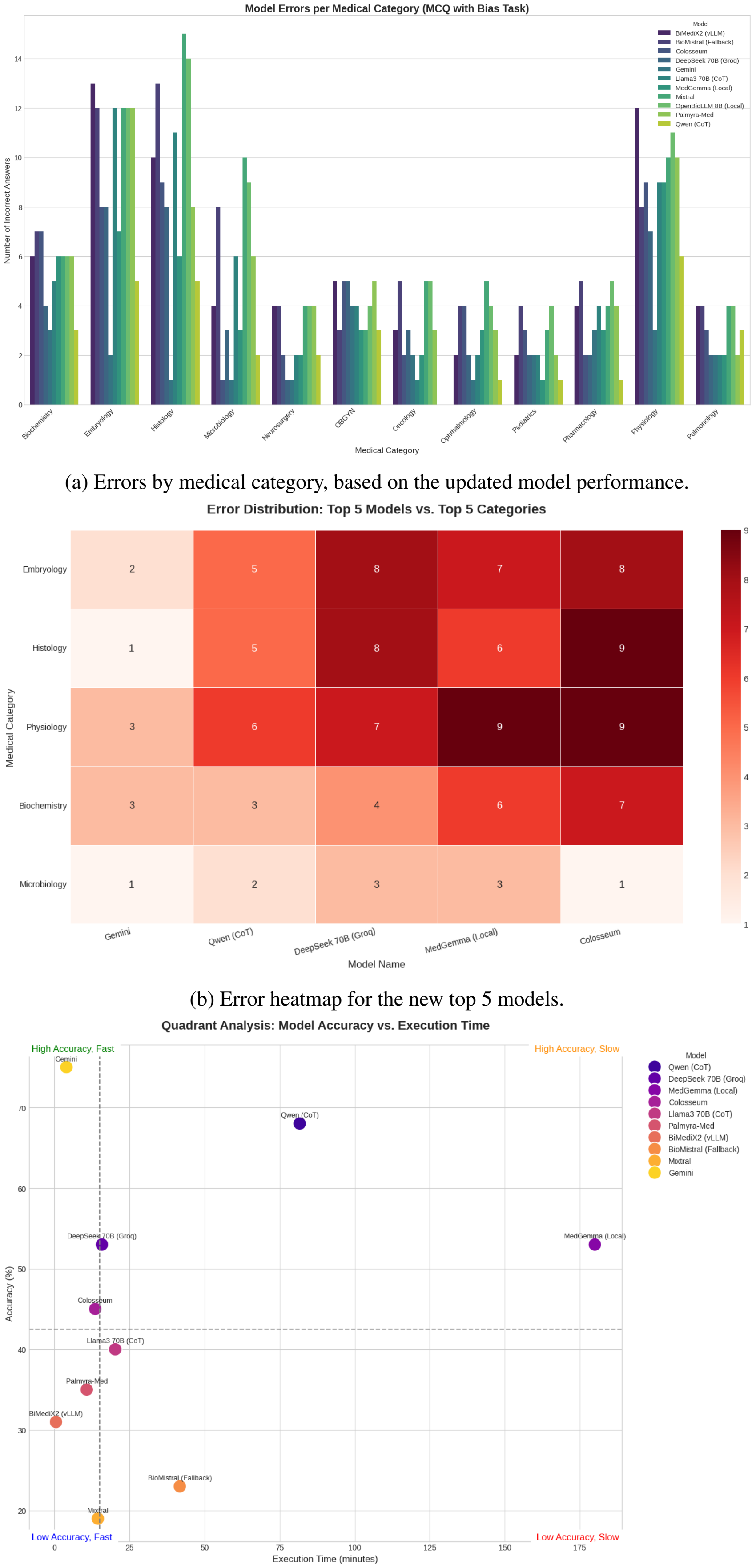


(a) Errors by medical category.

(b) Error heatmap for top models.

(c) Accuracy vs. Execution Time.

Figure 3: Detailed performance analysis for the Fill-in-the-Blank (FITB) task.

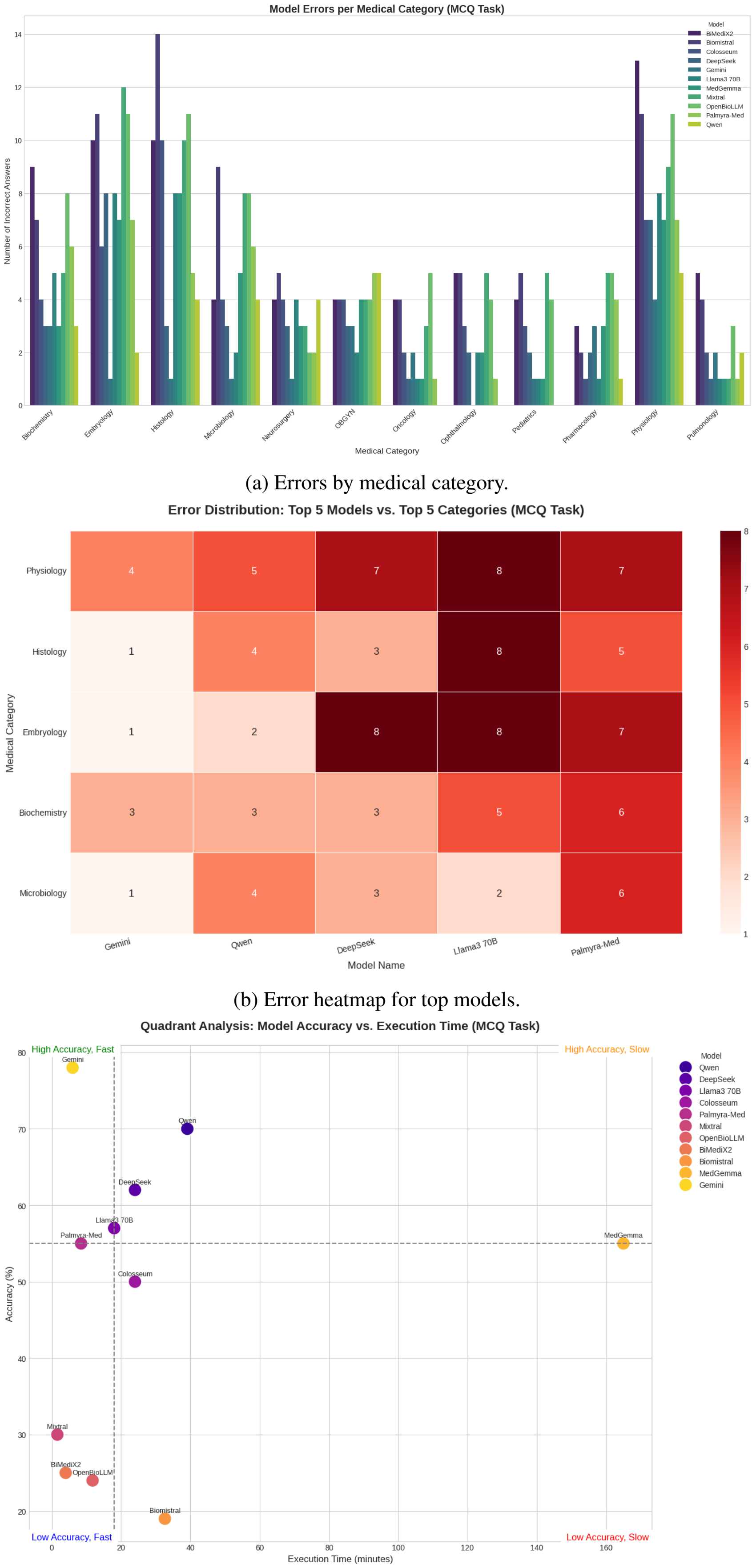


(a) Errors by medical category, based on the updated model performance.

(b) Error heatmap for the new top 5 models.

(c) Accuracy vs. Execution Time, including Gemini.

Figure 4: Detailed performance analysis for the Multiple Choice with Bias (MCQ w/ Bias) task using the latest data.



(a) Errors by medical category.

(b) Error heatmap for top models.

(c) Accuracy vs. Execution Time.

Figure 5: Detailed performance analysis for the standard Multiple Choice Question (MCQ) task.

Sakinah-AI at MentalQA: A Comparative Study of Few-Shot, Optimized, and Ensemble Methods for Arabic Mental Health Question Classification

Fatimah Emad Eldin, Mumina Abukar
Cairo University, The University of South Wales
{12422024441586@pg.cu.edu.eg, 74108361@students.southwales.ac.uk}

Abstract

This paper details the system developed by team Sakinah-AI for the MentalQA 2025 shared task, focusing on Arabic mental health question classification. We compare few-shot learning with Large Language Models against fine-tuning of BERT-based models (CAMeL-BERT and AraBERTv2). Few-shot learning with Palmyra-Med-70B achieved the highest weighted F1-score of 0.605, followed by hyperparameter-optimized CAMeL-BERT at 0.597. Notably, 5-fold ensemble methods proved detrimental to performance. Our results demonstrate that for low-resource specialized domains, both few-shot learning and optimized fine-tuning of appropriate base models outperform ensemble strategies. To ensure reproducibility all experimental code and final fine-tuned models are made publicly available.

1 Introduction

Arabic mental health NLP faces unique challenges due to limited annotated data and the linguistic complexity of user-generated content on mental health platforms. To address these challenges, we participated in the MentalQA 2025 shared task (Alhuzali et al., 2024), conducting a systematic comparison of three paradigms for Arabic mental health question classification: few-shot learning with large language models, optimized fine-tuning, and ensemble methods.

Our comparative study reveals critical insights for low-resource specialized domains. Few-shot learning with Palmyra-Med-70B (Kamble and Alshikh, 2023) achieved optimal performance (0.605 weighted F1-score), closely followed by hyperparameter-optimized CAMeL-BERT (0.597). Notably, CAMeL-BERT significantly outperformed AraBERTv2 (0.543), while k-fold ensemble methods proved detrimental to both models' performance. These findings challenge conventional wisdom that ensemble methods universally improve classification accuracy.

The results demonstrate that for small, specialized datasets, strategic model selection and optimization outweigh complex ensembling strategies. Domain-specific pre-training (Palmyra-Med) and careful hyperparameter tuning emerge as more effective approaches than aggregating multiple weak learners. To ensure reproducibility and facilitate future research, we provide open access to all experimental code and fine-tuned models via GitHub[1] and Hugging Face[2].

2 Background and Related Work

2.1 Task Overview and Dataset

The MentalQA 2025 shared task (Alhuzali et al., 2025) focuses on multi-label classification of Arabic mental health questions into seven categories: Diagnosis, Treatment, Anatomy/Physiology, Epidemiology, Healthy Lifestyle, Provider Choices, and Other. We participated in Track 1, Sub-Task 1, using a dataset of 500 annotated question-answer pairs (300 training, 50 development, 150 test) from Arabic mental health platforms characterized by informal, dialect-rich language.

2.2 Arabic Mental Health NLP Evolution

Early foundational work by Alghamdi et al. (2020) created the Arabic psychological forum corpus "Nafsany" and compared lexicon-based approaches against traditional machine learning models. Alasmari (2025) revealed a clear paradigm shift: pre-2022 studies relied on traditional machine learning and lexicon-based methods, while post-2022 research shifted towards transformer-based models like AraBERT (Antoun et al., 2020) and MARBERT, which consistently outperform traditional approaches.

[1] https://github.com/astral-fate/MentalQA2025/

[2] https://huggingface.co/collections/FatimahEmadEldin/sakinah-ai-at-mentalqa-689b2d707791cea458e97aaf

Alhuzali and Alasmari (2025) conducted comprehensive evaluation of Arabic PLMs on the MentalQA dataset, demonstrating that fine-tuned MARBERT achieved superior performance with Jaccard scores of 0.80 for question classification and 0.86 for answer classification, while few-shot learning with GPT-3.5 showed significant improvements over zero-shot approaches. Recent LLM evaluations by Zahran et al. (2025) across eight models on diverse Arabic mental health datasets found that prompt design is critical and few-shot techniques consistently improve performance. Practical applications include the "MindWave" app by Bensalah et al. (2024), which leverages AI for bilingual mental health support.

2.3 Research Gaps and Contribution

Despite progress, gaps remain: limited comparative studies between fine-tuning and few-shot approaches in Arabic mental health domains, insufficient evaluation of ensemble methods versus optimized single models in low-resource settings, and lack of systematic analysis comparing domain-specific versus general-purpose LLMs. Our work addresses these gaps by providing direct comparative evaluation between fine-tuning BERT-based models (CAMeL-BERT and AraBERTv2) and few-shot learning with large language models, systematically evaluating ensemble strategies against optimized single models in the low-resource MentalQA 2025 shared task setting.

3 Methodology

3.1 System Overview

Our system comprises two parallel pipelines for multi-label Arabic mental health question classification: Fine-Tuning and Few-Shot Learning (Figure 1). This design enables direct comparison between traditional supervised learning and contemporary in-context learning paradigms.

3.2 Fine-Tuning Pipeline

3.2.1 Base Model Selection

We selected two Arabic BERT variants with complementary strengths:

CAMeL-BERT-DA-Sentiment (Inoue et al., 2021): A specialized variant fine-tuned for sentiment analysis on Arabic dialectal text. We hypothesized its exposure to user-generated content would benefit processing informal mental health questions.

AraBERTv2 (Antoun et al., 2020): A widely-adopted baseline model for Arabic NLP tasks, providing robust comparison benchmarks.

3.2.2 Training Strategies

Optimized Single Models: We employed Optuna framework for automated hyperparameter optimization, systematically exploring learning rates (le-5 to 5e-5), batch sizes (8, 16), and epochs (10-20) to identify optimal configurations. The final hyperparameters used for the CAMEL-BERT model are detailed in Appendix B (Table 5).

K-Fold Ensembles: We trained five models using stratified cross-validation and averaged their predictions. This approach tests whether model diversity improves performance in low-resource settings.

3.2.3 Model Selection Rationale

We selected models to test three factors: domain specialization, architecture, and scale. Palmyra-Med-70B (Kamble and Alshikh, 2023) provides medical domain expertise. Mixtral-8x22B uses mixture-of-experts architecture, while Qwen3-235B represents dense transformers. Gpt-Oss-20B tests the lower performance boundary (20B parameters), and Colosseum-355B tests the upper boundary (355B parameters). This design isolates whether domain knowledge, architectural differences, or parameter scaling most impacts Arabic mental health classification. All models support Arabic and are accessible via NVIDIA NIM API.

3.3 Few-Shot Learning Pipeline

3.3.1 Model Selection Rationale

We selected models testing domain specialization (Palmyra-Med-70B), architecture (Mixtral-8x22B mixture-of-experts vs. Qwen3-235B dense transformer), and scale boundaries (Gpt-Oss-20B at 20B, Colosseum-355B at 355B parameters). All models support Arabic and are accessible via NVIDIA NIM API.

3.3.2 Prompt Engineering

We constructed structured prompts with: (1) explicit multi-label task instructions, (2) Arabic category definitions and examples, and (3) 3-5 diverse training exemplars. Models were explicitly instructed to "select ALL applicable categories" with multi-label demonstrations.

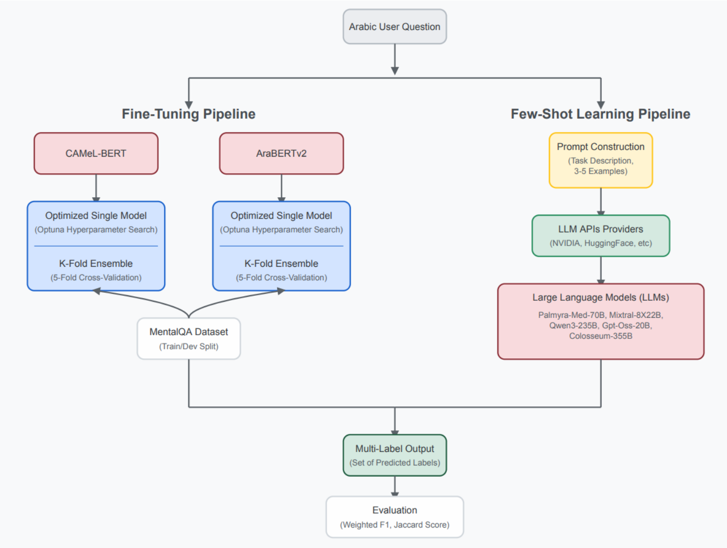


Figure 1: The Sakinah-AI System Architecture, illustrating two parallel processing pipelines.

3.4 Experimental Design

Our study follows a controlled comparison framework. For fine-tuning, we used 300 training samples with 50-sample development sets for hyperparameter optimization. For ensembles, we combined training and development sets (350 samples) for 5-fold cross-validation. Few-shot experiments used 3-5 training examples as in-context demonstrations. This design enables fair comparison across paradigms while addressing the low-resource constraints typical of specialized Arabic NLP domains.

4 Experimental Setup

4.1 Comparative Analysis Framework

We conduct a systematic comparison of three paradigms for Arabic mental health question classification: optimized fine-tuning, few-shot learning, and ensemble methods. This controlled evaluation addresses a critical research question: which approach performs best in low-resource specialized domains where traditional assumptions about ensemble superiority may not hold.

4.2 Data Configuration

The 500-sample dataset was partitioned into 300 training, 50 development, and 150 test samples. While this small size presents overfitting risks typical of specialized domains, we implement several mitigation strategies:

Fine-Tuning Protocol: Training set for model optimization, development set for hyperparameter selection, with early stopping based on development performance.

Ensemble Strategy: Combined training/development sets (350 samples) for stratified 5-fold cross-validation to maximize training data while maintaining validation integrity.

Few-Shot Design: Minimal training exposure (3-5 examples) inherently reduces overfitting risk while testing generalization from limited demonstrations. All final evaluations use the held-out test set to ensure unbiased performance estimates.

4.3 Evaluation Metrics

The primary evaluation metric is the weighted F1-score, which accounts for label imbalance (Sokolova and Lapalme, 2009). We additionally consider the Jaccard Score for multi-label evaluation (Manning et al., 2008).

Weighted F1-Score For a set of labels L, the weighted F1-score is calculated as:

$$\text{Weighted F1} = \sum_{l \in L} w_l \cdot F1_l \tag{1}$$

where w_l represents the proportion of instances of label l in the dataset, and $F1_l$ denotes the F1-score for that label, calculated as:

$$F1_l = 2 \cdot \frac{\text{Precision}_l \cdot \text{Recall}_l}{\text{Precision}_l + \text{Recall}_l} \tag{2}$$

Jaccard Score For individual predictions, where Y_{true} represents the set of true labels and Y_{pred} represents the set of predicted labels, the Jaccard score is:

$$J(Y_{\text{true}}, Y_{\text{pred}}) = \frac{|Y_{\text{true}} \cap Y_{\text{pred}}|}{|Y_{\text{true}} \cup Y_{\text{pred}}|} \quad (3)$$

The overall score represents the average Jaccard score across all samples.

5 Results

Our evaluation, conducted on the blind test set, reveals a distinct performance hierarchy among the different modeling paradigms. As shown in Table 1, the few-shot approach with a domain-specific LLM (Palmyra-Med-70B) achieved the highest weighted F1-score of 0.605. Closely following was the single, hyperparameter-optimized fine-tuned model, CAMeL-BERT (Opt.), with a score of 0.597. These top performers significantly outpaced all other models, particularly the ensemble variants, which consistently underperformed their single-model counterparts.

5.1 Error Analysis and Performance Patterns

To better understand these results, we conducted a detailed error analysis for both fine-tuned and few-shot models. A comprehensive quantitative and qualitative breakdown of model performance is available in Appendix C.

5.1.1 Fine-Tuned Model Analysis

As detailed in Table 2, the optimized CAMeL-BERT model maintains the lowest error counts across most categories, confirming its robustness. In contrast, the AraBERTv2 ensemble suffered a catastrophic performance collapse, with error counts surging in categories like **Anatomy and Physiology** (140 errors), **Other** (147 errors), and **Provider Choices** (122 errors). This pattern suggests that for smaller, specialized datasets, ensembling can amplify systematic model biases rather than mitigate variance, leading to degraded performance.

5.1.2 LLM Performance and Multi-Label Challenges

The error analysis for LLMs (Table 3) shows that Palmyra-Med-70B maintained a more balanced error profile compared to other models, which struggled significantly in high-support categories like **Diagnosis** and **Treatment**. A critical qualitative finding was the LLMs' systematic failure to adhere to multi-label instructions. Our prompt engineering (detailed in Appendix A Table 4) was specifically designed to prevent this by including: (1) explicit instructions to "perform precise multi-label classification" and "select ALL applicable categories," (2) clear examples of multi-label outputs (e.g., "Final Answer: A,D"), and (3) a structured format. Despite these safeguards, all tested LLMs frequently defaulted to predicting only a single label, even for questions where multiple categories were clearly relevant. This suggests a fundamental limitation in current instruction-following capabilities for complex classification tasks, possibly stemming from strong priors developed during pre-training on predominantly single-output tasks. This limitation likely suppressed the overall performance of all LLMs in our study.

5.2 Key Insights from Comparative Analysis

Domain Expertise vs. General Capability. The superior performance of Palmyra-Med-70B (0.605) over the much larger, general-purpose Qwen3-235B (0.325) highlights the profound value of domain-specific pre-training. Palmyra-Med's focused medical knowledge provided a decisive advantage in correctly interpreting the nuanced language of mental health questions, demonstrating that for specialized tasks, domain expertise can be more critical than model scale alone.

The Failure of Ensemble Methods. The consistent underperformance of k-fold ensembles challenges the conventional wisdom that they universally improve model robustness. For CAMeL-BERT, the ensemble F1-score (0.537) was notably lower than the optimized single model (0.597). The degradation was even more severe for AraBERTv2 (0.328 vs. 0.543). This outcome suggests that in low-resource settings, where individual models are trained on limited and potentially noisy data, they may develop high bias. In such cases, ensembling methods like averaging predictions can amplify these shared systematic errors rather than reducing variance, ultimately harming overall performance.

6 Discussion

Our results yield several key insights for specialized, low-resource domains. The superior performance of Palmyra-Med-70B (0.605) and optimized CAMEL-BERT (0.597) demonstrates that domain-specific pre-training and strategic single-model optimization are more effective than ensembling for Arabic mental health question classification. The consistent failure of our k-fold ensembles challenges the conventional wisdom that they univer-

Fine-Tuning	
Model Name	**Weighted F1-Score**
CAMeL-BERT (Optimized)	**0.597**
AraBERTv2 (Optimized)	0.543
CAMeL-BERT (K-Fold Ensemble)	0.537
AraBERTv2 (K-Fold Ensemble)	0.328

(a) Fine-Tuning Models

Few-Shot Learning	
Model Name	**Weighted F1-Score**
Palmyra-Med-70B	**0.605**
Mixtral-8X22B	0.563
Qwen3-235B	0.325
Gpt-Oss-20B	0.147
Colosseum-355B	0.014

(b) Few-Shot Learning Models

Table 1: Final results on the test set, comparing fine-tuned models against few-shot learning with LLMs. Optimized single models and domain-specific LLMs demonstrate superior performance.

Category	CAMeL-BERT		AraBERTv2	
	Opt.	Ens.	Opt.	Ens.
Anatomy	31	**18**	11	140
Diagnosis	**55**	71	53	65
Epidemiology	96	**85**	39	55
Lifestyle	**57**	102	44	38
Other	**3**	52	3	147
Provider	**31**	76	6	122
Treatment	**66**	**66**	63	85

Table 2: Error counts per category for all fine-tuned models. Lower values indicate better performance. Errors are calculated as Support × (1 - Recall).

Category	Palmyra	Mixtral	Qwen3	GPT-Oss	Colosseum
Anatomy	20	17	**10**	12	**10**
Diagnosis	**49**	52	67	74	84
Epidemiology	49	42	**37**	40	35
Lifestyle	**36**	38	39	39	37
Other	**3**	5	5	**3**	**3**
Provider	11	**9**	6	7	6
Treatment	50	**49**	66	81	85

Table 3: Error counts per category for few-shot LLMs. Lower values indicate better performance.

sally reduce errors. From a bias-variance perspective, ensembles are most effective at reducing variance by averaging the uncorrelated errors of diverse base learners. However, in low-resource settings with a small and specialized dataset, this core assumption is violated. The models trained on different folds of the data are not sufficiently diverse; instead, they learn similar systematic biases from the limited data. Consequently, the ensemble averages and reinforces these shared biases rather than canceling out random errors, leading to a notable degradation in performance, as seen with both CAMEL-BERT and AraBERTv2. While this study operated within the constraints of the provided dataset, future work could address these data limitations through several mitigation strategies. Data augmentation techniques, such as back-translation or contextual synonym replacement tailored to Arabic dialects, could create novel training instances. Furthermore, semi-supervised learning approaches could be employed to leverage vast amounts of unlabeled, in-domain text. By training a model on the existing labeled data and using it to generate pseudo-labels for unlabeled data, the training set could be effectively and cheaply expanded. A final significant finding was the LLMs' systematic failure to adhere to multi-label instructions despite explicit prompting, highlighting fundamental limitations in current instruction-following capabilities.

7 Conclusion

This paper presented the Sakinah-AI system for the MentalQA 2025 shared task, comparing few-shot learning, optimized fine-tuning, and ensemble methods for Arabic mental health question classification. Our results show that a domain-specific LLM, Palmyra-Med-70B, achieved the highest weighted F1-score (0.605), closely followed by an optimized CAMEL-BERT model (0.597). Notably, ensemble methods were detrimental to performance in this low-resource setting. The primary limitations of our study include the LLMs' difficulties with multi-label adherence and the small size of the training dataset. Furthermore. Future assessments must incorporate crucial dimensions such as clinical relevance and safety considerations to prevent harmful or inaccurate outputs. Moreover, focusing on model interpretability will be essential to build trust and utility for clinicians and end-users. Future work should explore advanced prompt engineering and data augmentation techniques while embedding these human-centered principles into the evaluation process.

Acknowledgments

We thank the organizers of the MentalQA 2025 shared task for their support and assistance in providing the dataset, evaluation framework, and coordination that made this research possible.

References

Ashwag Alasmari. 2025. A scoping review of arabic natural language processing for mental health. *Healthcare*, 13(9):963.

Norah S. Alghamdi, Hanan A. H. Mahmoud, Ajith Abraham, Samar A. Alanazi, and Laura García-Hernández. 2020. Predicting depression symptoms in an arabic psychological forum. *IEEE Access*, 8:57317–57334.

Hassan Alhuzali, Aseel Alasmari, and Hajar Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*.

Hassan Alhuzali and Ashwag Alasmari. 2025. Pre-trained language models for mental health: An empirical study on arabic q&a classification. *Healthcare*, 13(9):985.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, and 1 others. 2025. AraHealthQA 2025 shared task description paper. *arXiv preprint arXiv:2508.20047*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools*, pages 9–15, Marseille, France. European Language Resources Association.

Nouhaila Bensalah, Habib Ayad, Abdellah Adib, and Abdelhamid Ibn El Farouk. 2024. Mindwave app: Leveraging ai for mental health support in english and arabic. In *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, Marrakech, Morocco. IEEE.

Go Inoue, Bashar Al-Khafaji, 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 15–28.

Kiran Kamble and Waseem Alshikh. 2023. Palmyra-med: Instruction-based fine-tuning of llms enhancing medical domain performance. Preprint available on ResearchGate. Under review.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Noureldin Zahran, Aya E. Fouda, Radwa J. Hanafy, and Mohammed E. Fouda. 2025. A comprehensive evaluation of large language models on mental illnesses in arabic context. arXiv preprint. *Preprint*, arXiv:2501.06859.

A Few-Shots examples

The prompt used for all Large Language Model (LLM) evaluations was engineered to facilitate precise multi-label classification for Arabic mental health questions. As detailed in Table 4, the prompt architecture consists of four key components: a system prompt establishing an expert persona, a complete list of all seven categories with definitions, two diverse few-shot examples demonstrating the reasoning process and required multi-label output format (e.g., "Final Answer: A,D"), and a final task instruction for the target question. This structure was explicitly designed to guide the models in selecting all applicable categories and to counteract the observed tendency of LLMs to default to single-label outputs.

B Fine-Tuning Hyperparameters

The fine-tuning of the `CAMeL's` `bert-base-arabic-camelbert-mix-sentiment` model was conducted using the hyperparameters detailed in Table 5. These settings were configured using the Hugging Face Transformers library.

C Detailed Performance Analysis

This appendix provides a detailed quantitative and qualitative analysis of the top-performing models, based on the output from the error analysis script.

C.1 Quantitative Performance Summary

The table below summarizes the key performance metrics for the selected models. Palmyra-Med-70B demonstrates the best overall performance, closely followed by the optimized single model, CAMEL-BERT. The AraBERTv2-Ensemble model shows a significant degradation in performance across all metrics.

C.2 Per-Category F1-Score Matrix

To understand model performance on a more granular level, the following matrix presents the F1-score for each of the seven classification categories.

Component	Content
System Prompt	You are an expert in classifying Arabic patient questions into mental health categories. Perform precise multi-label classification.
Category List	**(A) Diagnosis:** Interpreting symptoms. **(B) Treatment:** Seeking therapies or medications. **(C) Anatomy and Physiology:** Basic medical knowledge. **(D) Epidemiology:** Course, prognosis, causes of diseases. **(E) Healthy Lifestyle:** Diet, exercise, mood control. **(F) Provider Choices:** Recommendations for doctors. **(Z) Other:** Does not fit other categories.
Example 1	**Question:** هل يعتبر الخوف من عدم الإنجاب مستقبلاً حالة عادية خاصةً لما أكون متعلقة بأطفال كثيراً وأنا على وشك الزواج أنا خائفة جداً **Reasoning:** The user is asking if their fear (a symptom) is normal and is concerned about its future course (prognosis). This fits 'Diagnosis' (interpreting a symptom) and 'Epidemiology' (prognosis). **Final Answer:** A,D
Example 2	**Question:** من سنة تقريباً وأنا أؤذي نفسي بأكثر من طريقة وما أعرف كيف أتخلص من هذه العادة، وبدأت تجيني أفكار بإنهاء حياتي وحاولت أنتحر بأكثر من مرة وأكثر من طريقة **Reasoning:** The user describes self-harm and suicidal thoughts and is asking how to get rid of this habit. This is a clear call for 'Treatment' (seeking therapy/help) and relates to 'Healthy Lifestyle' (self-help, mood control). **Final Answer:** B,E
Task	Classify the following question. Provide your reasoning and then the final answer. **Question:** {Target Question} **Reasoning:** **Final Answer:**

Table 4: Structure and content of the few-shot prompt used for LLM inference.

Hyperparameter	Value
Model & Tokenizer	
Base Model	CAMeL-BERT (mix-sentiment)
Max Sequence Length	256
Training Arguments	
Epochs	15
Batch Size	8
Gradient Accum. Steps	2
Learning Rate	2e-5
Warmup Steps	100
Weight Decay	0.01
Optimizer	AdamW
FP16 Precision	True
Loss Function	
Loss Type	Focal Loss
Alpha (α)	1.0
Gamma (γ)	2.0

Table 5: Hyperparameters for the optimized fine-tuning of CAMeL-BERT.

Metric	Palmyra-Med-70B	CAMEL-BERT Opt	AraBERTv2 Ens.
Exact Match Ratio	12.67%	11.33%	0.00%
Macro Jaccard Score	0.2623	0.2445	0.1115
Weighted F1-Score	0.60	0.59	0.26

Table 6: Overall performance metrics on the blind test set.

Both Palmyra-Med and CAMEL-BERT perform strongly on high-support categories like Diagnosis (A) and Treatment (B), while the Ensemble model fails completely on Treatment and Healthy Lifestyle questions.

C.3 Error Analysis Matrix

The following examples from the test set illustrate common failure modes for different models, highlighting the challenges of multi-label classification and the pitfalls of ensembling in low-resource settings.

Category	Palmyra-Med-70B	CAMEL-BERT Opt	AraBERTv2 Ens.
(A) Diagnosis	0.75	0.76	0.71
(B) Treatment	**0.74**	0.70	0.00
(C) Anatomy/Phys.	0.09	**0.15**	0.12
(D) Epidemiology	**0.44**	0.37	0.18
(E) Healthy Lifestyle	0.38	**0.41**	0.00
(F) Provider Choices	**0.15**	0.00	0.09
(Z) Other	0.00	0.00	**0.04**

Table 7: Per-category F1-scores for each model. Higher is better.

Error Type	Question & Analysis	Labels
Multi-Label Failure (Palmyra-Med-70B)	**Question:** بكاء مفاجى ،حزن ، فقدان الوزن بدون سبب، الاكتئاب، عدم الثقة بالنفس، القلق ، التوتر ، الانطوائية، **Analysis:** The user lists numerous symptoms ('A'), is implicitly asking for a solution ('B'), and is concerned about the course of the illness ('D'). The model correctly identifies 'Diagnosis' but fails to capture the other required labels.	**True:** A, B, D **Predicted:** A
Ensemble Hallucination (AraBERTv2 Ensemble)	**Question:** ماهو افضل دواء منوم وذاا تأثير سريع وقوي لاني اعاني من ارق ولا استطيع النوم ابداا الرجاء الاجابه؟؟ **Analysis:** A direct question about medication ('B'). The ensemble model not only misses the correct label entirely but also hallucinates four incorrect and irrelevant labels, demonstrating a catastrophic failure.	**True:** B **Predicted:** A, C, F, Z
Domain Specialization (CAMEL-BERT Opt)	**Question:** رجاء أريد معلومات عن دواء جديد اسمه ايشل پلنځپرن شكرا **Analysis:** This is a clear request for information about a specific treatment ('B'). The optimized CAMEL-BERT model, attuned to user-generated dialectal content, correctly classifies this. The log shows that the baseline AraBERT model failed to produce any prediction for this item, highlighting the robustness of the optimized model.	**True:** B **Predicted:** B

Table 8: Illustrative examples of misclassification cases from the test set.

MindLLM at AraHealthQA 2025 Track 1: Leveraging Large Language Models for Mental Health Question Answering

Nejood Abdulaziz Bin Eshaq
Department of Computer Science, King Khalid University, Abha 62521, Saudi Arabia
446818545@kku.edu.sa

Abstract

This paper presents our submission to the AraHealthQA 2025 shared task (Alhuzali et al., 2025), Sub-task 3: Arabic Mental Health Question Answering. We evaluated four large language models—GPT-4o, Gemini, Allam, and Qwen—using various prompting strategies. A simple 3-shot prompt, instructing the model to respond in Arabic, consistently outperformed zero-shot, 5-shot, and more complex methods. GPT-4o achieved the best results, with a **BERTScore F1 of 0.670** on the official hidden test set, ranking **2nd overall**. The system required no fine-tuning or external data, relying solely on prompt design and consistent evaluation.

1 Introduction

Mental health disorders, such as obsessive-compulsive disorder (OCD), depression, and suicidal ideation, affect millions worldwide, significantly impairing well-being and daily functioning (World Health Organization, 2022). Early intervention can enhance recovery, prevent severe outcomes like self-harm, and reduce the broader societal and economic burden (Patel et al., 2018). Moreover, prioritizing mental health care helps break stigma and encourages individuals to seek the support they need. The AraHealthQA 2025 shared task (Alhuzali et al., 2025) addresses the growing demand for accessible and culturally appropriate mental health resources for Arabic-speaking populations. It highlights both the social importance of providing trustworthy support and the technical challenges posed by modeling Arabic psychological discourse. The shared task comprises three subtasks; our work focuses on Subtask 3: Question Answering, which requires generating accurate, informative, and empathetic answers to mental health questions written in Arabic.

We experimented with four large language models (LLMs)—GPT-4o (OpenAI, 2024), Gemini (Team et al., 2023), Allam (Bari et al., 2024), and Qwen (Qwen Team, 2024)—and explored multiple prompting strategies, including zero-shot, few-shot, chain-of-thought (Wei et al., 2022), and self-consistency (Wang et al., 2022). Prompt selection was conducted using Meta's *LLaMA-3-70B-Instruct model (8192-token context)* as an LLM-as-a-judge, evaluated via BERTScore F1 (Zhang et al., 2019). After iterative testing, we adopted a 3-shot prompting approach and selected GPT-4o as our final submission model, based on its alignment with expert-written answers.

Our system achieved 2nd place in the official leaderboard with a BERTScore F1 of 0.670. Key challenges during development included ensuring clean and well-structured input data, enforcing consistent and controlled answer formats, and handling ambiguous or emotionally sensitive queries that require careful phrasing to avoid misinterpretation, especially in a mental health context where psychological state and cultural background may influence understanding.

The full code and evaluation scripts are available at: https://github.com/njoudae/AraHealthQA_2025_subtasck_3/tree/main.

2 Related Work

Recent years have seen significant progress in Arabic NLP for mental health, although challenges like limited data and cultural complexities still hinder its development. (Alasmari, 2025) offers a scoping review that outlines the current state of Arabic NLP in mental health, covering methods from classical machine learning models like SVM and Random Forest to more advanced transformer models such as AraBERT and MARBERT. The review notes a strong focus on detecting depression and suicidal tendencies, often leveraging social media data, and sheds light on both the strengths and drawbacks of existing techniques. While transformer models

Proceedings of The Third Arabic Natural Language Processing Conference, pages 149–154
November 8-9, 2025 ©2025 Association for Computational Linguistics

have delivered impressive results, the study emphasizes the lack of dataset variety and the urgent need for culturally aware tools that accommodate dialectal differences and address societal stigma in Arabic-speaking regions.

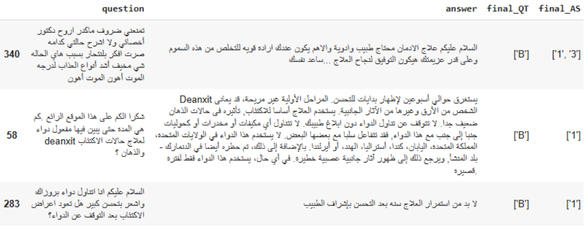
Expanding on this groundwork, (Alhuzali and Alasmari, 2025) carried out a practical assessment of pre-trained language models (PLMs) for classifying Arabic mental health Q&A using the MentalQA dataset. They compared traditional machine learning techniques, Arabic PLMs like MARBERT and CAMeLBERT, and prompt-based approaches using GPT-3.5/4. Their findings revealed that PLMs significantly outperformed older feature-based models, with MARBERT delivering the best results. Interestingly, GPT-3.5 prompt-based methods excelled in few-shot learning situations, showing promise for applications in low-resource languages. However, the study also highlighted a critical limitation: the small size of the MentalQA dataset (only 500 samples), which impacts how broadly the findings can be applied.

Shifting the focus to real-world applications, (Bensalah et al., 2024) introduced MindWave, a bilingual Arabic–English mental health support app. The system uses NLP and sentiment analysis on both text and voice inputs to identify signs of burnout and depression. To tackle the shortage of Arabic sentiment datasets, the researchers built a large parallel English–Arabic medical corpus containing 945,000 sentences. They then fine-tuned machine translation models to develop classifiers tailored to Arabic. Additionally, the study compared various Arabic tokenization techniques, offering useful insights into best practices. Unlike previous efforts that focused mainly on classification or Q&A tasks, MindWave showcases how NLP tools can be seamlessly integrated into interactive support platforms and communities.

Lastly, (Zahran et al., 2025) performed a wide-ranging evaluation of large language models (LLMs) in the context of Arabic mental health. This study stands out as one of the first to deeply assess how well LLMs function in this domain. The authors pointed out both the benefits and risks of LLMs: while these models can generate meaningful and relevant responses, concerns about empathy, cultural appropriateness, and safety persist. Compared to more specialized PLMs, general-purpose LLMs showed inconsistent reliability, reinforcing the need for domain-specific adaptation and human monitoring. Collectively, these studies highlight the importance of building richer datasets, adopting multifaceted evaluation methods (beyond basic accuracy scores like BERTScore), and developing culturally sensitive NLP tools. Our research builds on these findings by focusing on prompt-based evaluation within the AraHealthQA framework, tackling both performance and ethical dimensions in this underexplored area.

3 Task and Dataset Description

The AraHealthQA 2025 shared task (Alhuzali et al., 2025) provides a benchmark dataset for evaluating Arabic mental health question answering systems. The dataset, MentalQA, was recently accepted in IEEE ACCESS and consists of 500 annotated samples of real user-submitted psychological questions and expert-written answers in Arabic (Alhuzali et al., 2024).



	question	answer	final_QT	final_AS
340	تمنعني ضروف ماكدر اروح دكتور أخصائي ولا اشرح حالتي كدامه صرت افكر بلنتحار بسبب هاي الحاله شي مخيف أشد أنواع العذاب لدرجه الموت أهون الموت أهون	السلام عليكم علاج الادمان محتاج طبيب وادوية والاهم يكون عندك اراده قويه للتخلص من هذه السموم وعلى قدر عزيمتك هيكون التوفيق لنجاح العلاج ...ساعد نفسك	['B']	['1', '3']
58	شكرا الكم على هذا الموقع الرائع .كم هي المده حتى يبين فيها مفعول دواء deanxit لعلاج حالات الاكتئاب والذهان ؟	يستغرق حوالي أسبوعين لإظهار بدايات للتحسن. المراحل الأولية غير مريحة، قد يعاني Deanxit الشخص من الأرق وغيرها من الآثار الجانبية. يستخدم العلاج أساسا للاكتئاب, تأثيره في حالات الذهان ضعيف جدا. لا تتوقف عن تناول الدواء دون ابلاغ طبيبك. لا تتناول أي مكيفات أو مخدرات أو كحوليات جنبا إلى جنب مع هذا الدواء, فقد تتفاعل سلبا مع بعضها البعض. لا يستخدم هذا الدواء في الولايات المتحدة، المملكة المتحدة، اليابان، كندا، أستراليا، الهند، أو أيرلندا. بالإضافة إلى ذلك، تم حظره أيضا في الدنمارك - بلد المنشأ, ويرجع ذلك إلى ظهور آثار جانبية عصبية خطيرة. في أي حال، يستخدم هذا الدواء فقط لفترة قصيرة.	['B']	['1']
283	السلام عليكم انا اتناول دواء بروزاك واشعر بتحسن كبير هل تعود اعراض الاكتئاب بعد التوقف عن الدواء؟	لا بد من استمرار العلاج سنه بعد التحسن بإشراف الطبيب	['B']	['1']

Figure 1: Data samples from MentalQA.

We participated in Sub-task 3: Question Answering, which requires generating expert-level answers to Arabic mental health questions. This task builds on the earlier classification sub-tasks and aims to develop systems capable of providing accurate and useful responses. The official evaluation metric used for Sub-task 3 is BERTScore (F1).

While recent studies have begun to explore Arabic NLP for mental health, prior work has primarily focused on resource creation, small-scale evaluations, or application-level prototypes. Building on these efforts, our contribution is to systematically evaluate multiple large language models on the AraHealthQA dataset and to analyze differences in response quality and their alignment with expert-written answers in the Arabic psychological domain.

4 System Description

Our system follows a structured prompt-based generation workflow using pre-trained large language models (LLMs) without any fine-tuning. The process which consists of four stages: (1) data preparation, (2) prompt design, (3) model setup, and (4) evaluation, was provided in Appendix Figure 2

4.1 Data Preparation

We used the AraHealthQA Subtask 3 dataset, which contains 350 samples for training and development, and 150 samples for testing. All samples were kept in Arabic to preserve cultural and linguistic nuances. The dataset was cleaned, and minor inconsistencies were corrected to ensure reliability, and example selection ensured topical diversity and cultural appropriateness.

4.2 Prompt Design & Strategies

Prompts were designed using real question–answer pairs from the dataset. We experimented with:

- Zero-shot
- Few-shot (3-shot, 5-shot)
- Chain-of-thought (CoT) (Wei et al., 2022)
- Self-consistency (Wang et al., 2022)
- Ensemble refinement

Zero-shot achieved a BERTScore F1 of 0.61, while 3-shot improved to 0.66. Self-consistency with 3-shot produced stable results, but 5-shot and CoT slightly degraded performance. Ensemble refinement did not improve scores.

4.3 Model Setup

The final configuration fixed the 3-shot prompt format across all models. No external data beyond the provided samples were used. Models included:

- GPT-4o (OpenAI, 2024)
- Gemini (Team et al., 2023)
- Allam (Bari et al., 2024)
- Qwen (Qwen Team, 2024)

Models were accessed via public APIs or Hugging Face, and all runs used fixed seeds for reproducibility.

4.4 Evaluation

For each test question, a 3-shot prompt was dynamically constructed. Model outputs were compared against expert-written answers using BERTScore F1 (Zhang et al., 2019). GPT-4o achieved the highest balance between accuracy and empathy, Gemini was empathetic but less precise, Allam favored technical terminology, and Qwen tended toward generic responses.

5 Experimental Setup

Data Split Usage

For Subtask 3, the organizers released 350 annotated samples for training and development, and 150 samples as a hidden test set (Table 1). Each entry contains: (1) the question, (2) the expert-written answer, (3) the question type, and (4) the answer strategy. Question types include *diagnosis*, *treatment*, *epidemiology*, and *healthy lifestyle*, while answer strategies are *informational*, *direct guidance*, and *emotional support*.

Table 1: MentalQA dataset distribution for Subtask 3.

	Train/Dev	Test	Total
Samples	350	150	500

From the training split, we selected 10 representative question–answer pairs covering all question types and answer strategies to construct prompting examples. These examples were fixed and reused across all prompting strategies to ensure fair comparisons. Final evaluation was conducted on the entire hidden test set.

External Tools and Libraries

All models were used in their original form without fine-tuning:

- **GPT-4o** and **Gemini**: accessed via their official APIs (accessed on 20 July 2025).
- **Allam** and **Qwen**: accessed via Hugging Face Inference API (accessed on 20 July 2025).
- **LLaMA-3-70B-Instruct (Grattafiori et al., 2024)**: accessed via Groq API (Groq, 2024) for prompt evaluation (accessed on 20 July 2025).

Table 2 summarizes the full prompting configurations used for each model.

Model	Temp.	Top-p	Max tokens
GPT-4o	0.1	0.9	1024
Gemini 2.5	0.1	0.9	1024
ALLaM-7B	0.1	0.9	1024
Qwen2.5-7B	0.1	0.9	1024
LLaMA-3 70B	0.1	0.9	1024

Table 2: Prompting parameters used across models.

Key libraries

- Hugging Face Hub version: 0.34.3
- BERTScore v0.3.11
- openai v0.28
- Google Generative AI version: 0.8.5
- Python 3.11.13

Evaluation Metric

We used BERTScore F1 (Zhang et al., 2019) with the multilingual model to compare system outputs against expert-written answers. Scores were computed using the official `bert_score` implementation (v0.3.11) with default multilingual settings for Arabic. This metric measures semantic similarity between generated answers and references, accounting for lexical and contextual matches.

Detailed results and prompt strategy that used are shown in Appendix Figure 3

6 Results

Our final system, which used GPT-4o with 3-shot prompting, achieved a BERTScore F1 of 0.67 on the official test set and was ranked 2nd overall in Sub-task 3 of the AraHealthQA 2025 shared task (Alhuzali et al., 2025).

The full results of model comparisons and prompting strategies are presented in Appendix Table 4 and Table 3

Table 3: BERTScore F1 performance of different LLMs on the official train set (3-shot prompting).

Model	BERTScore F1
GPT-4o	0.6551
Allam	0.6316
Gemini	0.6210
Qwen	0.6131

Table 4: BERTScore F1 performance across different prompting strategies, evaluated using LLaMA-3-70B-Instruct.

Prompting Strategy	BERTScore F1
Zero-shot	0.6100
3-shot	0.6600
3-shot + self-consistency	0.6600
Few-shot (5-shot)	0.6400
Chain-of-thought	0.6150
3-shot + ensemble refinement	0.6100

LLaMA-3-70B-Instruct was used only as a reference model to compare prompting strategies (Table 4) and was not included in Table 3, since our leaderboard submission relied on other models.

In the development phase, we conducted extensive ablation studies to compare various prompting strategies across multiple models. 3-shot prompting consistently outperformed zero-shot, 5-shot, and more complex techniques such as chain-of-thought reasoning, self-consistency, and ensemble refinement. While chain-of-thought prompting introduced more structured reasoning, it slightly decreased performance on BERTScore metrics. Increasing to 5-shot did not yield additional benefit and often produced redundant outputs. As a result, 3-shot prompting was selected for its superior performance and simplicity.

In the development phase, we conducted extensive ablation studies to compare various prompting strategies across multiple models. 3-shot prompting consistently outperformed zero-shot, 5-shot, and more complex techniques such as chain-of-thought reasoning, self-consistency, and ensemble refinement. While chain-of-thought prompting introduced more structured reasoning, it slightly decreased performance on BERTScore metrics. Increasing to 5-shot did not yield additional benefits and often produced redundant outputs. As a result, 3-shot prompting was selected for its superior performance and simplicity. No major hallucinations or foreign-language artifacts were observed in the generated answers. Notably, the selected model (GPT-4o) avoided making explicit diagnostic claims or recommending specific medical treatments. Instead, the system provided general guidance, informative responses, and help-seeking suggestions — a desirable behavior for mental health applications where only qualified professionals should deliver clinical diagnoses or therapeutic interventions. This aligns well with the task's goal of producing educational and supportive content without overstepping ethical boundaries. All reported results are based on the official submission. No post-submission modifications or evaluations were performed.

All reported results are based on the official submission. No post-submission modifications or evaluations were performed.

7 Limitations

The dataset is relatively limited in size, which restricts the ability to generalize the findings. As a result, there's a need to expand the database in the future. While the BERTScore F1 serves as a useful metric for quantitative assessment, relying solely on it falls short of capturing critical elements such as empathy, safety, and cultural nuances. To address this, we plan to implement a more holistic set of evaluation standards moving forward. These will encompass emotional factors, health relevance, contextual appropriateness, harm prevention, and risk awareness. We aim to combine the LLM-as-a-Judge framework with human judgment to produce outcomes that are both more trustworthy and grounded in real-world considerations.

8 Conclusion and future work

In this work, we presented a prompt-based question answering system for Arabic mental health queries, developed as part of the AraHealthQA 2025 shared task. Our final system was built on GPT-4o using 3-shot prompting with carefully selected examples from the training data. The system demonstrated the ability to generate coherent, informative, and non-diagnostic responses that were consistent with the expert-written reference answers provided in the dataset.

For future work, we plan to explore fine-tuning Arabic LLMs on the full dataset to enhance contextual alignment, as well as investigate retrieval-augmented generation (RAG) techniques to integrate external knowledge sources and improve factual accuracy in complex queries. We also intend to involve mental health professionals in the evaluation process to assess the psychological appropriateness and safety of model-generated answers.

9 Acknowledgments

We thank the organizers of the AraHealthQA 2025 shared task for providing the dataset and evaluation platform.

References

Ashwag Alasmari. 2025. A scoping review of arabic natural language processing for mental health. In *Healthcare*, volume 13, page 963. MDPI.

Hassan Alhuzali and Ashwag Alasmari. 2025. Pre-trained language models for mental health: An empirical study on arabic q&a classification. In *Healthcare*, volume 13, page 985. MDPI.

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash, and Leen Kharouf. 2025. Arahealthqa 2025 shared task description paper. In *Proceedings of ArabicNLP 2025*.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan Al-Rashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, and 1 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Nouhaila Bensalah, Habib Ayad, Abdellah Adib, and Abdelhamid Ibn El Farouk. 2024. Mindwave app: Leveraging ai for mental health support in english and arabic. In *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 1–6. IEEE.

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, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Inc. Groq. 2024. Meta Llama 3 70b (8192) served via groq api. Accessed: 2025-07-20.

OpenAI. 2024. GPT-4o: Openai's omnimodal model. Accessed: 2025-07-25.

Vikram Patel, Shekhar Saxena, Crick Lund, Graham Thornicroft, Florence Baingana, Paul Bolton, Dan Chisholm, Pamela Y. Collins, Janice L. Cooper, Julian Eaton, Helen Herrman, Mazen M. Herzallah, Yu Huang, Mark J. D. Jordans, Arthur Kleinman, María Elena Medina-Mora, Graham Morgan, Unaiza Niaz, Oye Gureje Omigbodun, and 9 others. 2018. The lancet commission on global mental health and sustainable development. *The Lancet*, 392(10157):1553–1598.

Qwen Team. 2024. Qwen2.5: A party of foundation models. Accessed: 2025-07-25.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

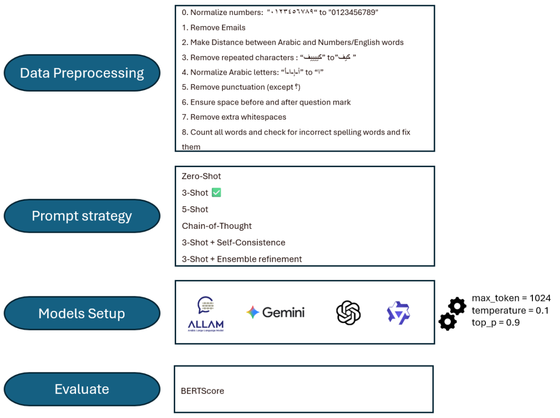
Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

World Health Organization. 2022. World mental health report: Transforming mental health for all. Accessed: 2025-07-02.

Noureldin Zahran, Aya E Fouda, Radwa J Hanafy, and Mohammed E Fouda. 2025. A comprehensive evaluation of large language models on mental illnesses in arabic context. *arXiv preprint arXiv:2501.06859*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Appendix

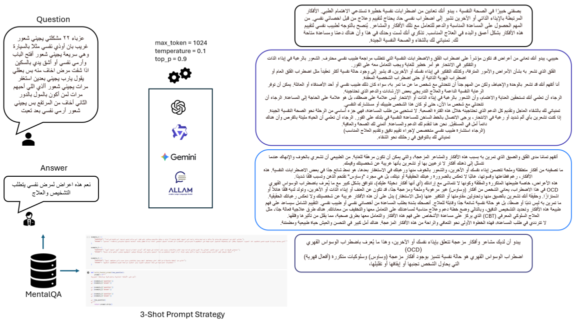


0. Normalize numbers: "٠١٢٣٤٥٦٧٨٩" to "0123456789"
1. Remove Emails
2. Make Distance between Arabic and Numbers/English words
3. Remove repeated characters : "كيييف" to"كيف "
4. Normalize Arabic letters: "أ-إ-ا-آ" to "ا"
5. Remove punctuation (except ؟)
6. Ensure space before and after question mark
7. Remove extra whitespaces
8. Count all words and check for incorrect spelling words and fix them

Zero-Shot
3-Shot ✅
5-Shot
Chain-of-Thought
3-Shot + Self-Consistence
3-Shot + Ensemble refinement

Figure 2: System pipeline

B Appendix



عزباء ٢٢ مشكلتي يجيني شعور
غريب بان أوذي نفسي مثلا بالسيارة
وهي سريعة يجيني شعور أفتح الباب
وأرمي نفسي أو أشق يدي بالسكين
اذا شفت مرض اخاف منه بس بعقلي
يقول يارب يجيني بعدين استغفر
مرات يجيني شعور أاذي اللي أحبهم
مرات لمن أكون بالمول بالدور
الثاني أخاف من المرتفع بس يجيني
شعور أرمي نفسي بعد تعبت

نعم هذه اعراض لمرض نفسي يتطلب
التشخيص والعلاج

بصفتي خبيرًا في الصحة النفسية ، يبدو أنك تعانين من اضطرابات نفسية خطيرة تستدعي الاهتمام الطبي. الأفكار
المرتبطة بالإيذاء الذاتي أو الآخرين تشير إلى اضطراب نفسي حاد يحتاج لتقييم وعلاج من قبل اخصائي نفسي. من
المهم الحصول على المساعدة المناسبة والدعم للتعامل مع تلك الأفكار والمشاعر. يُنصح بالتوجه لطبيب نفسي لتقييم
هذه الأفكار بشكل أعمق والبدء في العلاج المناسب. تذكري أنك لست وحدك في هذا وأن هناك دعمًا ومساعدة متاحة
لك. تمنياتي لك بالشفاء والصحة النفسية الجيدة.

حبيبي، يبدو أنك تعاني من أعراض قد تكون مؤشراً على اضطراب القلق والاضطرابات النفسية التي تتطلب مراجعة طبيب نفسي محترف. الشعور بالرغبة في إيذاء الذات
والتفكير في الانتحار هو أمر خطير للغاية ويجب التعامل معه على الفور.
القلق الذي تشعر به بشأن الأمراض والأمور المترفة، وكذلك التفكير في إيذاء نفسك أو الآخرين، قد يشير إلى وجود حالة نفسية أكثر تعقيداً مثل اضطراب القلق العام أو
اضطراب الهوية الذاتية أو حتى اضطراب الشخصية المعقدة.
أنا أتفهم أنك قد تشعر بالوحدة والإحباط، ولكن من المهم جداً أن تتحدثي مع شخص ما عن ما تمر به، سواء كان ذلك طبيب نفسي أو أحد الأصدقاء أو العائلة. يمكن أن توفر
الرعاية النفسية الداعمة والعلاج التدريجي بعض الإرشادات والدعم الذي تحتاجينه.
الرجاء أن تعلمي أنك تستحقين العناية والاهتمام، وأن الشعور بالرغبة في إيذاء الذات أو الانتحار ليس علامة على ضعفك، بل هو علامة على الحاجة إلى المساعدة. الرجاء أن
تتحدثي مع شخص ما الآن، حتى لو كان هذا الشخص طبيبك أو مستشارك النفسي.
تمنياتي لك بالشفاء العاجل وتقديم كل الدعم الذي تحتاجينه خلال هذه الفترة الصعبة. لا تستحيي من طلب المساعدة، فهي جزء أساسي من الرحلة نحو الصحة النفسية الجيدة.
إذا كنت تشعرين بأي ألم شديد أو رغبة في الانتحار، يرجى الاتصال بالخط الساخن للمساعدة النفسية في بلدك على الفور. الرجاء أن تعلمي أن الحياة مليئة بالفرص وأن هناك
دائماً أمل في المستقبل. نحن هنا لنقدم لك الدعم والمساعدة. أتمنى لك الصحة والعافية.
(الرجاء استشارة طبيب نفسي متخصص لإجراء تقييم دقيق وتقديم العلاج المناسب)
تمنياتي لك بالتوفيق في رحلتك نحو الشفاء.

أتفهم تمامًا مدى القلق والضيق الذي تمرين به بسبب هذه الأفكار والمشاعر المزعجة، والتي يمكن أن تكون مرهقة للغاية. من الطبيعي أن تشعري بالخوف والإنهاك عندما
تتسلل إلى ذهنك أفكار لا ترغبين بها أو تشعرين بأنها غريبة عن شخصيتك وقيمك.
ما تصفينه من أفكار متطفلة وملحة تتضمن إيذاء نفسك أو الآخرين، والشعور بالخوف منها ورغبتك في الاستغفار بعدها، هو نمط شائع جدًا في بعض الاضطرابات النفسية. هذه
الأفكار، رغم فظاعتها وقسوتها، غالبًا لا تعكس بالضرورة رغبتك الحقيقية أو نيتك، بل هي مجرد "وساوس" تقتحم الذهن وتسبب قلقًا شديدًا.
هذه الأعراض، خاصة طبيعتها المتكررة والمقلقة وكونها لا تتماشى مع إرادتك (أي أنها أفكار دخيلة عليك)، تتوافق بشكل كبير مع ما يُعرف باضطراب الوسواس القهري
OCD في هذا الاضطراب، يعاني الشخص من أفكار (وساوس) غير مرغوبة وملحة ومزعجة جدًا، قد تكون عن العنف أو إيذاء الذات أو الآخرين، وتولد لديه قلقًا هائلاً أو
اشمئزازًا. وحقيقة أنك تشعرين بالضيق منها وتحاولين مقاومتها أو التكفير عنها (مثل الاستغفار) يدل على أن هذه الأفكار غريبة عن شخصيتك ولا تعكس رغباتك الحقيقية.
ما تمرين به ليس ذنبًا أو ضعفًا، بل هو حالة نفسية شائعة جدًا وقابلة للعلاج. أنصحك بشدة بطلب المساعدة من أخصائي نفسي أو طبيب نفسي. التقييم الشامل سيساعد على فهم
طبيعة هذه الأفكار وتحديد التشخيص الدقيق، وبالتالي وضع خطة دعم وعلاج مناسبة لمساعدتك على التعامل معها والتخفيف من معاناتك. هناك طرق علاجية فعالة جدًا، مثل
العلاج السلوكي المعرفي (CBT) الذي يركز على مساعدة الأشخاص على فهم هذه الأفكار والتعامل معها بطرق صحية، مما يقلل من تأثيرها وقلقها.
لا تترددي في طلب المساعدة، فهذه الخطوة الأولى نحو التعافي والراحة من هذه الأفكار المزعجة. هناك أمل كبير في التحسن والعيش حياة طبيعية ومطمئنة.

يبدو أن لديك مشاعر وأفكار مزعجة تتعلق بإيذاء نفسك أو الآخرين، وهذا ما يُعرف باضطراب الوسواس القهري
(OCD)
اضطراب الوسواس القهري هو حالة نفسية تتميز بوجود أفكار مزعجة (وساوس) وسلوكيات متكررة (أفعال قهرية)
التي يحاول الشخص تجنبها أو إيقافها أو تقليلها.

Figure 3: An illustrative example from the MentalQA dataset showing the question, gold reference, and generated answers using prompting strategies (3-shot).

C Appendix

All implementation details, including full prompt examples and evaluation scripts, are available in our GitHub repository: https://github.com/njoudae/AraHealthQA_2025_subtasck_3/tree/main.

Quasar at AraHealthQA Track 1 : Leveraging Zero-Shot Large Language Models for Question and Answer Categorization in Arabic Mental Health

Adiba Fairooz Chowdhury and **MD Sagor Chowdhury**
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2004014, u2004010}@student.cuet.ac.bd

Abstract

Pre-trained language models (PLMs) show potential for advancing mental health care, yet their effectiveness in Arabic mental health contexts is underexplored. This study evaluates PLMs on two multi-label classification tasks from the AraHealthQA 2025 shared task Track 1: question categorization and answer strategy classification. We systematically evaluate several LLMs spanning Arabic-specialized, multilingual, and general-purpose architectures using zero-shot inference, with comparative analysis revealing Qwen3-14B's superior performance. Our approach combines prompt-based inference, label mapping, and strategically crafted Arabic prompts. Experiments on 350 training and 150 test samples demonstrate competitive performance, securing 4^{th} place in both tasks (Question F1: 0.52, Answer F1: 0.76; Question Jaccard: 0.41, Answer Jaccard: 0.66). These findings reveal strengths and limitations of current PLMs for detecting complex intents in Arabic mental health contexts.

1 Introduction

Pre-Trained Language Models (PLMs) have transformed many domains, including medicine (He et al., 2023), yet research on their application to mental health remains nascent. PLMs offer promising support for patients and tools for healthcare providers, from conversational agents (Liu et al., 2023; Brocki et al., 2023) to classifying user input for therapeutic intervention (Sharma et al., 2023). However, effective mental health PLMs must grasp symptom nuances and subjectivity, a greater challenge for Arabic. Spoken by over 400 million people, Arabic's rich morphology, dialect diversity, right-to-left script, and context-sensitive character shapes complicate NLP (Guellil et al., 2021). Despite advances in other languages (Atapattu et al., 2022; Kabir et al., 2022; Sun et al., 2021), Arabic mental health NLP is underexplored, with limited prior studies (Abdulsalam et al., 2024; Aldhafer and Yakhlef, 2022; Al-Musallam and Al-Abdullatif, 2022; Al-Laith and Alenezi, 2021; El-Ramly et al., 2021).

This paper reports our submission to AraHealthQA 2025 Track 1 (Alhuzali et al., 2025), which targets Arabic mental health discourse. We assess zero-shot performance of large PLMs, particularly Qwen3-14B, on multi-label Question Categorization and Answer Strategy Classification. Ranking 4th in both subtasks, our results show zero-shot PLMs can approach fine-tuned models in low-resource, culturally specific settings. This paper's main contributions are as follows:

- First prompt-based, zero-shot classification on MentalQA 2025 without fine-tuning.
- Culturally adapted Arabic prompts for mental health classification.
- Systematic evaluation demonstrating Qwen3-14B's competitive performance.
- Analysis of PLM strengths and limitations for Arabic mental health contexts.

Implementation details are available at[1].

2 Background

2.1 Task Description

ArahealthQA Track 1 is a shared task on Arabic mental health question answering, consisting of:

- **Sub-Task 1:** Multi-label Question Categorization[2] —classifying questions into predefined categories (Table 1).
- **Sub-Task 2:** Multi-label Answer Strategy Classification[3] — categorizing answers according to predefined strategies (Table 1).

[1] https://github.com/AdibAFC/Quasar_ArahealthQA-Track1-MentalQA
[2] https://www.codabench.org/competitions/8559/
[3] https://www.codabench.org/competitions/8730/

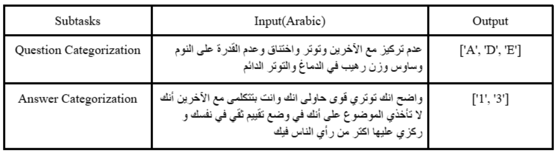
Proceedings of The Third Arabic Natural Language Processing Conference, pages 155–163
November 8-9, 2025 ©2025 Association for Computational Linguistics

#	Q-Types	#	A-Types
A	Diagnosis	1	Information
B	Treatment	2	Direct Guidance
C	Anatomy and Physiology	3	Emotional Support
D	Epidemiology		
E	Healthy Lifestyle		
F	Provider Choices		
Z	Other		

Table 1: Question (Q) and Answer (A) types.

2.2 Dataset

The shared task uses the MentalQA dataset (Alhuzali et al., 2024), containing 500 annotated Arabic Q&A posts (350 development, 150 test) specialized in mental health discourse. Table 2 illustrates input-output examples.



Subtasks	Input(Arabic)	Output
Question Categorization	عدم تركيز مع الآخرين وتوتر واختناق وعدم القدرة على النوم وساوس وزن رهيب في الدماغ والتوتر الدائم	['A', 'D', 'E']
Answer Categorization	واضح انك توتري قوى حاولى انك وانت بتتكلمى مع الآخرين أنك لا تأخذي الموضوع على أنك في وضع تقييم ثقي في نفسك و ركزي عليها اكتر من رأي الناس فيك	['1', '3']

Table 2: Sample input-output mapping with Arabic question-answer and corresponding labels

2.3 Related Work

PLM development for English has progressed rapidly with models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019). Despite Arabic being the fourth most prevalent language online with over 400 million speakers, few PLMs exist due to Arabic's linguistic complexity (Shaalan et al., 2019). Mental health NLP research has primarily focused on English, leaving Arabic question and answer classification underexplored. The recent MentalQA dataset marks important progress, with reviews emphasizing the need for specialized Arabic NLP resources in mental health (Alasmari, 2025). Recent efforts also include MedArabiQ, benchmarking large language models on Arabic medical tasks (Abu Daoud et al., 2025).

Recent developments in Arabic mental health NLP have shown promising advances (Alhuzali and Alasmari, 2025; Zahran et al., 2025), demonstrating both the effectiveness of domain-specific adaptations and the challenges of applying contemporary LLMs to Arabic mental health discourse. Practical applications have emerged (Bensalah et al., 2024), leveraging AI for multilingual mental health support. Comprehensive reviews (Alasmari, 2025) have systematically analyzed Arabic NLP applications in mental health, identifying key gaps and research directions.

This work provides novel benchmarks and insights for culturally aware, low-resource Arabic mental health NLP applications through large-scale multilingual PLMs and prompt-based adaptation.

3 System Overview

Our system evaluates multiple large language models for Arabic medical question classification using a unified zero-shot inference pipeline. We systematically compare six models, spanning Arabic-specialized, multilingual, and general-purpose architectures, to assess their effectiveness specifically in mental health discourse classification.

3.1 Model Selection Rationale

We selected models based on three criteria: (1) Arabic language capabilities, (2) architectural diversity (encoder-only vs decoder-only), and (3) computational feasibility. The Qwen family was chosen for demonstrated multilingual performance, Llama3.1 for its broad adoption and Arabic support, DeepSeek for its reasoning capabilities, and AraBERTv2 as the Arabic-specialized baseline.

3.2 Multi-Model Architecture Framework

Our evaluation framework accommodates diverse architectures, dividing them into generative (decoder-only) and classification (encoder-only) models. The generative models include Qwen3-14B[4] (14.8B parameters), Qwen2.5-7B[5], and Qwen2-7B[6], each with a 32K context length, Llama3.1-8B — Meta's instruction-tuned multilingual model[7], and DeepSeek R1-7B — a distilled model[8] optimized for reasoning tasks. On the classification side, we use AraBERTv2, an Arabic-specialized BERT variant (aubmindlab/bert-base-arabertv2[9]). To handle large models, we apply 4-bit NF4 quantization, which reduces memory usage by approximately 75% without compromising performance (Dettmers et al., 2021). Memory

[4] https://huggingface.co/Qwen/Qwen3-14B

[5] https://huggingface.co/unsloth/Qwen2.5-7B-Instruct

[6] https://huggingface.co/Qwen/Qwen2-7B

[7] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[8] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B

[9] https://huggingface.co/aubmindlab/bert-base-arabertv2

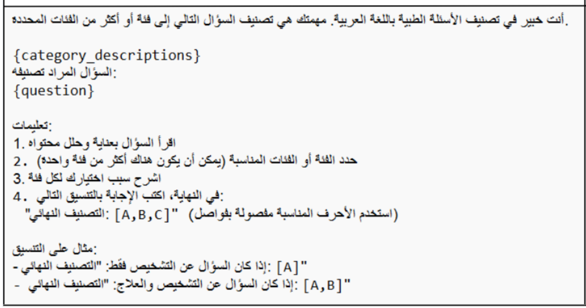
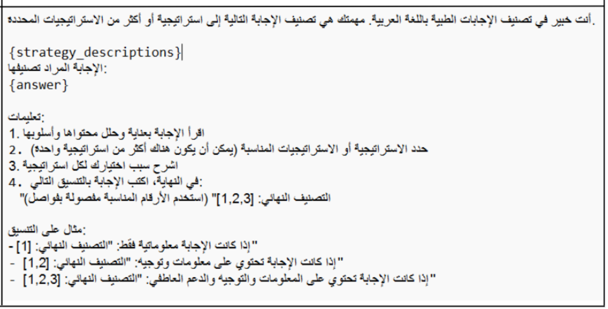
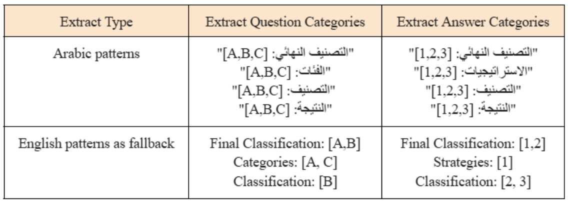
Prompt Template for Question Classification	Prompt Template for Answer Classification
 أنت خبير في تصنيف الأسئلة الطبية باللغة العربية. مهمتك هي تصنيف السؤال التالي إلى فئة أو أكثر من الفئات المحدده. {category_descriptions} السؤال المراد تصنيفه: {question} تعليمات: 1. اقرأ السؤال بعناية وحلل محتواه 2. حدد الفئة أو الفئات المناسبة (يمكن أن يكون هناك أكثر من فئة واحده) 3. اشرح سبب اختيارك لكل فئة 4. في النهاية، اكتب الإجابة بالتنسيق التالي: "التصنيف النهائي: [A,B,C]" (استخدم الأحرف المناسبة مفصولة بفواصل) مثال على التنسيق: - إذا كان السؤال عن التشخيص فقط: "التصنيف النهائي: [A]" - إذا كان السؤال عن التشخيص والعلاج: "التصنيف النهائي: [A,B]"	 أنت خبير في تصنيف الإجابات الطبية باللغة العربية. مهمتك هي تصنيف الإجابة التالية إلى استراتيجية أو أكثر من الاستراتيجيات المحدده. {strategy_descriptions} الإجابة المراد تصنيفها: {answer} تعليمات: 1. اقرأ الإجابة بعناية وحلل محتواها وأسلوبها 2. حدد الاستراتيجية أو الاستراتيجيات المناسبة (يمكن أن يكون هناك أكثر من استراتيجية واحده) 3. اشرح سبب اختيارك لكل استراتيجية 4. في النهاية، اكتب الإجابة بالتنسيق التالي: "التصنيف النهائي: [1,2,3]" (استخدم الأرقام المناسبة مفصولة بفواصل) مثال على التنسيق: - إذا كانت الإجابة معلوماتية فقط: "التصنيف النهائي: [1]" - إذا كانت الإجابة تحتوي على معلومات وتوجيه: "التصنيف النهائي: [1,2]" - إذا كانت الإجابة تحتوي على المعلومات والتوجيه والدعم العاطفي: "التصنيف النهائي: [1,2,3]"

Table 3: Structured prompt templates for Arabic question classification (left) and answer classification (right).

optimization is further achieved through dynamic GPU memory management, and the entire system is implemented within the unified Hugging Face ecosystem (further details are in Appendix A).

3.3 Methodology

Our approach ensures consistent evaluation protocols across all models using task-specific Arabic prompts designed for cross-model compatibility. These prompts include structured category listings and reasoning instructions, as summarized in Table 3. We apply model-specific adaptations such as enabling thinking_mode=True in Qwen models to facilitate structured reasoning, while other generative models use standard chat templates with equivalent reasoning prompts. For BERT-based models, classification heads are employed with prompt-based input formatting. Outputs from all models undergo a robust regex-based extraction process capable of handling multilingual responses effectively, as illustrated in Figure 1.



Extract Type	Extract Question Categories	Extract Answer Categories
Arabic patterns	"التصنيف النهائي: [A,B,C]" "الفئات: [A,B,C]" "التصنيف: [A,B,C]" "النتيجة: [A,B,C]"	"التصنيف النهائي: [1,2,3]" "الاستراتيجيات: [1,2,3]" "التصنيف: [1,2,3]" "النتيجة: [1,2,3]"
English patterns as fallback	Final Classification: [A,B] Categories: [A, C] Classification: [B]	Final Classification: [1,2] Strategies: [1] Classification: [2, 3]

Figure 1: Regex-based pattern recognition process for extracting categories from Arabic and English responses

3.4 System Pipeline and Algorithm

Our system employs a structured zero-shot classification pipeline supporting both generative and classification models under a unified framework. As illustrated in Figure 2, it uses task-specific Arabic prompts with structured reasoning and model-specific strategies like thinking mode to ensure consistent classification of medical questions. Outputs are standardized through a robust regex-based multi-pattern label extraction process, enabling direct comparison among Arabic-specialized, multilingual, and general-purpose models within the same system.

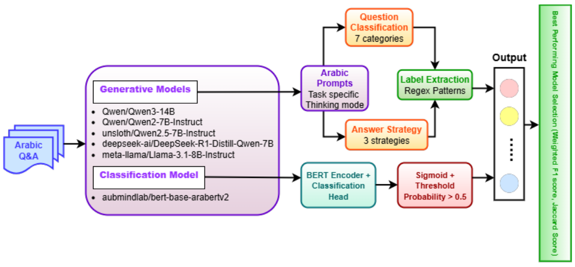

Arabic Q&A
Generative Models
Qwen/Qwen3-14B
Qwen/Qwen2-7B-Instruct
unsloth/Qwen2.5-7B-Instruct
deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
meta-llama/Llama-3.1-8B-Instruct
Classification Model
aubmindlab/bert-base-arabertv2
Arabic Prompts
Task specific
Thinking mode
Question Classification
7 categories
Answer Strategy
3 strategies
Label Extraction
Regex Patterns
BERT Encoder + Classification Head
Sigmoid + Threshold Probability > 0.5
Output
Best Performing Model Selection (Weighted F1 score, Jaccard Score)

Figure 2: Zero-shot Arabic medical classification pipeline supporting multiple LLM architectures with unified prompt engineering and evaluation framework

3.5 Technical Challenges and Solutions

Achieving consistent Arabic understanding across diverse architectures was a key challenge. Our framework supports both encoder-only models like AraBERTv2 and decoder-only generative models such as Qwen, Llama, and DeepSeek, enabling direct comparison. Dynamic prompt engineering and modular regex-based output processing ensure robustness across varied response formats and languages. Memory limitations were managed with adaptive quantization—4-bit NF4 for large models and standard precision for smaller ones. Evaluation uses probabilistic thresholds and macro-averaged F1 scores for standardized, fair assessment across all models.

3.6 System Example

Detailed examples of model classifications for both questions and answers are provided in Appendix B, Figures 5 and 6.

4 Experimental Setup

4.1 Data Usage and Implementation

The model is used in a zero-shot setting without fine-tuning. Train_Dev.tsv (350 samples) was used for evaluation with gold-standard labels, while test.csv (150 samples) was used for blind inference. Arabic questions were processed without preprocessing to preserve semantic integrity. Prompts were constructed in Arabic with explicit multi-category classification instructions.

4.2 Evaluation Metrics

The model's performance on the labeled `Train_Dev.tsv` set was evaluated using the **Weighted F1-Score** and **Jaccard Similarity**.

$$F1_{\text{weighted}} = \frac{\sum_{i=1}^{n} w_i \cdot \frac{2 \cdot \text{precision}_i \cdot \text{recall}_i}{\text{precision}_i + \text{recall}_i}}{\sum_{i=1}^{n} w_i}$$

$$\text{Jaccard}(T_i, P_i) = \frac{|T_i \cap P_i|}{|T_i \cup P_i|}$$

where T_i and P_i are the ground-truth and predicted label vectors for sample i, w_i is the number of true instances of class i, and n is the total number of classes. Complete implementation details are provided in Appendix A.

5 Results

5.1 Development Set Evaluation

We report performance of various LLMs on both classification tasks using the labeled `Train_Dev.tsv` dataset in zero-shot setting.

Model	**Question Class.**		**Answer Class.**	
	F1-Score	**Jaccard**	**F1-Score**	**Jaccard**
Random Baseline	0.326	0.199	0.541	0.378
Majority Class	0.245	0.193	0.451	0.397
Weighted Random	0.386	0.250	0.587	0.432
Qwen3-14B	**0.507**	**0.363**	**0.767**	**0.628**
Qwen2.5-7B	0.504	0.356	0.693	0.529
Qwen2-7B	0.499	0.344	0.688	0.530
DeepSeek R1-7B	0.330	0.213	0.723	0.556
Llama3.1-8B	0.315	0.207	0.632	0.541
AraBERTv2	N/A	N/A	0.466	0.563

5.2 Official Competition Results

Our best-performing system (Qwen3-14B) achieved 4th place in both subtasks on the blind test set (150 samples):

- Question Classification: Weighted F1-Score = 0.52, Jaccard = 0.41
- Answer Classification: Weighted F1-Score = 0.76, Jaccard = 0.66

5.3 Comparative Analysis

Qwen3-14B consistently outperformed other models and baseline methods, with performance substantially exceeding random, weighted and majority class baselines. Complete baseline analysis and model comparisons are provided in Appendix C.

5.4 Error Analysis

Analysis of confusion matrices reveals key error patterns: Question classification shows frequent confusion between *Diagnosis* and *Healthy Lifestyle* (89 cases), and between *Treatment* and *Diagnosis* (111 cases). Answer classification shows significant confusion between *Information* vs. *Direct Guidance* categories. Technical issues included irregular formatting requiring robust regex post-processing and occasional model refusal to classify ambiguous content.

Technical Implementation Issues

- Irregular formatting requiring robust regex post-processing
- Inconsistent Arabic/English label mixing in model outputs
- Occasional model refusal to classify ambiguous mental health content

The confusion matrices (Figures 3 and 4) illustrate these classification patterns, with diagonal dominance indicating generally good performance despite the identified challenges. Specific examples of model output errors for both tasks are provided in Appendix D

5.5 Cross-Architecture Analysis

Our systematic evaluation reveals distinct performance patterns across model architectures:

Qwen Family Dominance: The Qwen models (Qwen3 > Qwen2.5 > Qwen2) demonstrate superior Arabic comprehension, with Qwen3-14B achieving the highest scores in both tasks. This suggests that the Qwen architecture's multilingual pretraining particularly benefits Arabic mental health discourse.

Model Size Effects: Larger models generally outperform smaller ones within the same

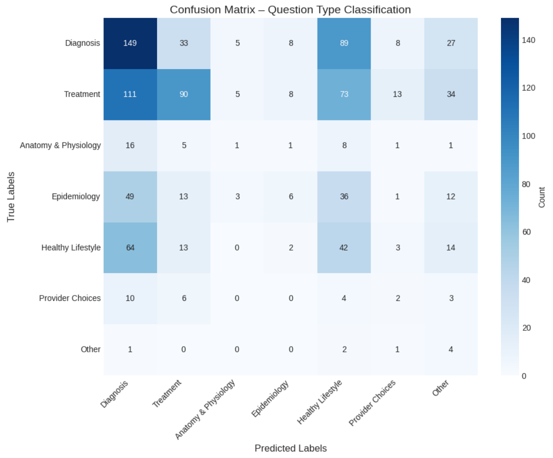


Figure 3: Question Classification Confusion Matrix

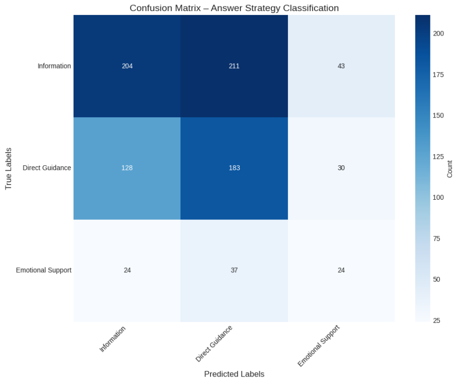


Figure 4: Answer Classification Confusion Matrix

family, with Qwen3-14B (14B) outperforming Qwen2.5-7B and Qwen2-7B in question classification, though the gap is smaller for answer classification.

Specialized vs General Models: The comparison between Arabic-specialized AraBERTv2 and multilingual generative models reveals that recent large multilingual models can match or exceed specialized models in domain-specific tasks.

6 Discussion

6.1 Model Architecture Insights

Our comparative analysis reveals several insights: (1) The Qwen family's superior performance suggests that certain multilingual pre-training strategies better capture Arabic linguistic nuances, (2) Decoder-only models generally outperform encoder-only models for these classification tasks, and (3) Model size provides diminishing returns within the same architecture family.

6.2 Arabic-Specific Challenges

The performance gap between models highlights the continued challenges in Arabic NLP, where models not specifically designed for Arabic underperform significantly (Llama3.1 vs Qwen3-14B: 0.315 vs 0.507 F1 for questions).

7 Conclusion

We presented a systematic evaluation of multiple LLM architectures for zero-shot Arabic mental health classification, with our best system (Qwen3-14B) achieving 4th place in both tasks. Our comparative analysis demonstrates that recent multilingual models can achieve competitive performance without fine-tuning, though significant performance gaps exist between model families. The Qwen architecture's superior performance suggests that specific multilingual pre-training strategies better capture Arabic linguistic nuances. Limitations include lack of domain-specific adaptation and output format variability across models. Future work includes domain-specific fine-tuning on larger Arabic medical corpora, incorporating retrieval-augmented generation for contextual understanding, evaluation across diverse Arabic dialects, investigating prompt engineering techniques for medical domains.

Acknowledgments

We thank all contributors who supported this work, especially the developers of the LLM models evaluated. We appreciate feedback from anonymous reviewers. This research did not receive specific funding.

References

Amani Abdulsalam, Abdullah Alhothali, and Saeed Al-Ghamdi. 2024. Detecting suicidality in arabic tweets using machine learning and deep learning techniques. *Arabian Journal for Science and Engineering*, 49:12729–12742.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E Shamout. 2025. Medarabiq: Benchmarking large language models on arabic medical tasks. *arXiv e-prints*, pages arXiv–2505.

Afnan Al-Laith and Mansour Alenezi. 2021. Monitoring people's emotions and symptoms from arabic tweets during the covid-19 pandemic. *Information*, 12(2):86.

Nouf Al-Musallam and Majed Al-Abdullatif. 2022. Depression detection through identifying depressive arabic tweets from saudi arabia: Machine learning approach. In *2022 Fifth National Conference of Saudi Computer Colleges (NCCC)*, pages 11–18. IEEE.

Ashwag Alasmari. 2025. A scoping review of arabic natural language processing for mental health. In *Healthcare*, volume 13-9, page 963. MDPI.

Saeed H. Aldhafer and Mahdi Yakhlef. 2022. Depression detection in arabic tweets using deep learning. In *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 1–6. IEEE.

Hassan Alhuzali and Ashwag Alasmari. 2025. Pretrained language models for mental health: An empirical study on arabic qa classification. *Healthcare*, 13(9).

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*, 12:101155–101165.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash, and Leen Kharouf. 2025. Arahealthqa 2025 shared task description paper. In *Proceedings of ArabicNLP 2025*.

Thilina Atapattu, Madurika Herath, Chiran Elvitigala, Prashan de Zoysa, Kasun Gunawardana, Madurika Thilakaratne, Kasun de Zoysa, and Katrina Falkner. 2022. Emoment: An emotion annotated mental health corpus from two south asian countries. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6991–7001.

Nour Bensalah, Hiba Ayad, Ali Adib, and Abdelouahid Ibn El Farouk. 2024. Mindwave app: Leveraging ai for mental health support in english and arabic. In *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 1–6. IEEE.

Laura Brocki, George C. Dyer, Agnieszka Gładka, and N.C. Chung. 2023. Deep learning mental health dialogue system. In *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 395–398. IEEE.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2021. 8-bit optimizers via block-wise quantization. *Preprint*, arXiv:2110.02861. ICLR 2022 Spotlight Version.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Mahmoud El-Ramly, Hadeer Abu-Elyazid, Yomna Mo'men, Ghadah Alshaer, Nermine Adib, Khaled A. Eldeen, and Manar El-Shazly. 2021. Cairodep: Detecting depression in arabic posts using bert transformers. In *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 207–212. IEEE.

Imane Guellil, Hichem Saâdane, Faycal Azouaou, Bachir Gueni, and Damien Nouvel. 2021. Arabic natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences*, 33:497–507.

Kuan He, Ruixiang Mao, Qian Lin, Yuxuan Ruan, Xiaodong Lan, Minlie Feng, and Erik Cambria. 2023. A survey of large language models for healthcare: From data, technology, and applications to accountability and ethics. *arXiv e-prints*, pages arXiv–2310.05694.

Md Kamrul Kabir, Md Islam, A N M Baki Kabir, Anwarul Haque, and Md Kamrul Rhaman. 2022. Detection of depression severity using bengali social media posts on mental health: Study using natural language processing techniques. *JMIR Formative Research*, 6:e36118.

Jiaming Liu, Dan Li, Hao Cao, Tao Ren, Zhen Liao, and Jian Wu. 2023. Chatcounselor: A large language model for mental health support. *arXiv e-prints*, pages arXiv–2309.15461.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Khaled Shaalan, Sajid Siddiqui, Moustafa Alkhatib, and Ahmed Abdel Monem. 2019. Challenges in arabic natural language processing. In *Computational Linguistics, Speech and Image Processing for Arabic Language*, pages 59–83. World Scientific.

Aniket Sharma, I.W. Lin, Adam S. Miner, David C. Atkins, and Tim Althoff. 2023. Human–ai collaboration enables more empathic conversations in text-based peer-to-peer mental health support. *Nature Machine Intelligence*, 5:46–57.

Huan Sun, Zheng Lin, Chuxu Zheng, Shasha Liu, and Minlie Huang. 2021. Psyqa: A chinese dataset for generating long counseling text for mental health support. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1489–1503.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Nourhan Zahran, Amr E Fouda, Rana J Hanafy, and Mostafa E Fouda. 2025. A comprehensive evaluation of large language models on mental illnesses in arabic context. *arXiv preprint arXiv:2501.06859*.

A Implementation Details

A.1 Technical Environment

- **Hardware:** NVIDIA A100 GPU (80GB VRAM), 128GB RAM
- **Software:** Python 3.10, CUDA 11.8, Transformers v4.51.0, PyTorch v2.2.0, BitsAndBytes v0.43.0

A.2 Multi-Model Configuration Parameters

Generative Models (Qwen, Llama, DeepSeek): Temperature: 0.7, max_new_tokens: 512, top_p: 0.9, repetition_penalty: 1.1. Model-specific adaptations: Qwen models use `thinking_mode=True`, DeepSeek uses temperature=0.3 for reasoning, Llama3.1 uses standard instruct templates.

Classification Model (AraBERTv2): Linear classification head (768 → num_labels), max sequence length: 512, full precision due to smaller size and architectural differences.

Quantization Strategy: 4-bit NF4 for models >10B parameters (Qwen3-14B), 8-bit or full precision for smaller models based on VRAM availability.

A.3 Architecture-Specific Implementation

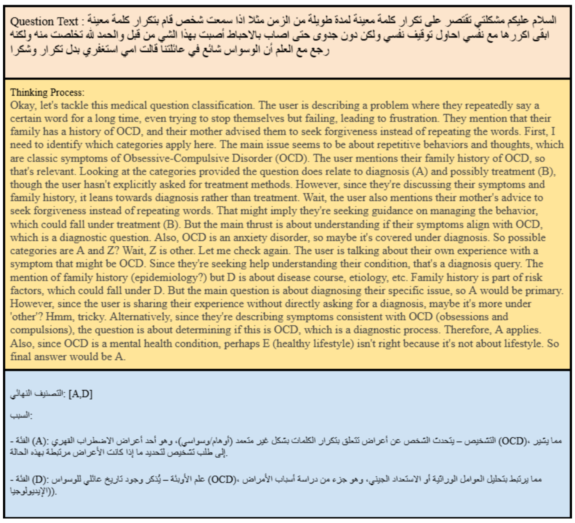
Decoder-Only Models: Unified generation pipeline with model-specific chat templates and reasoning prompts. Output processed via regex extraction and label mapping.

Encoder-Only Model (AraBERTv2): Direct classification using linear head with sigmoid activation for multi-label prediction. Compatible only with answer classification task due to formatting constraints.

A.4 Memory Management and Evaluation

Sequential model loading with dynamic quantization prevents OOM errors. Memory requirements: Qwen3-14B (4-bit): 8GB, 7B models (8-bit): 4-6GB, AraBERTv2: 1GB. All models evaluated using identical metrics (sklearn implementation) with `average=’weighted’` for fair comparison. .

B System Example Figures



Question Text : السلام عليكم مشكلتي تقتصر على تكرار كلمة معينة لمدة طويلة من الزمن مثلا اذا سمعت شخص قام بتكرار كلمة معينة ابقى اكررها مع نفسي احاول توقيف نفسي ولكن دون جدوى حتى اصاب بالاحباط أصبت بهذا الشي من قبل والحمد لله تخلصت منه ولكنه رجع مع العلم أن الوسواس شائع في عائلتنا قالت امي استغفري بدل تكرار وشكرا

Thinking Process:
Okay, let's tackle this medical question classification. The user is describing a problem where they repeatedly say a certain word for a long time, even trying to stop themselves but failing, leading to frustration. They mention that their family has a history of OCD, and their mother advised them to seek forgiveness instead of repeating the words. First, I need to identify which categories apply here. The main issue seems to be about repetitive behaviors and thoughts, which are classic symptoms of Obsessive-Compulsive Disorder (OCD). The user mentions their family history of OCD, so that's relevant. Looking at the categories provided the question does relate to diagnosis (A) and possibly treatment (B), though the user hasn't explicitly asked for treatment methods. However, since they're discussing their symptoms and family history, it leans towards diagnosis rather than treatment. Wait, the user also mentions their mother's advice to seek forgiveness instead of repeating words. That might imply they're seeking guidance on managing the behavior, which could fall under treatment (B). But the main thrust is about understanding if their symptoms align with OCD, which is a diagnostic question. Also, OCD is an anxiety disorder, so maybe it's covered under diagnosis. So possible categories are A and Z? Wait, Z is other. Let me check again. The user is talking about their own experience with a symptom that might be OCD. Since they're seeking help understanding their condition, that's a diagnosis query. The mention of family history (epidemiology?) but D is about disease course, etiology, etc. Family history is part of risk factors, which could fall under D. But the main question is about diagnosing their specific issue, so A would be primary. However, since the user is sharing their experience without directly asking for a diagnosis, maybe it's more under 'other'? Hmm, tricky. Alternatively, since they're describing symptoms consistent with OCD (obsessions and compulsions), the question is about determining if this is OCD, which is a diagnostic process. Therefore, A applies. Also, since OCD is a mental health condition, perhaps E (healthy lifestyle) isn't right because it's not about lifestyle. So final answer would be A.

التصنيف النهائي: [A,D]

السبب:

- الفئة (A): التشخيص – يتحدث الشخص عن أعراض تتعلق بتكرار الكلمات بشكل غير متعمد (أوهام/وسواسي)، وهو أحد أعراض الاضطراب القهري (OCD)، مما يشير إلى طلب تشخيص لتحديد ما إذا كانت الأعراض مرتبطة بهذه الحالة.

- الفئة (D): علم الأوبئة – يُذكر وجود تاريخ عائلي للوسواس (OCD)، مما يرتبط بتحليل العوامل الوراثية أو الاستعداد الجيني، وهو جزء من دراسة أسباب الأمراض (الإيبديولوجيا)).

Figure 5: This demonstrates classification of a question about repetitive behaviors and obsessive thoughts, related to OCD. The system analyzes that the person is describing symptoms consistent with OCD and seeking understanding rather than directly asking for diagnosis. It classifies this as Category A (Diagnosis) since the question involves determining if symptoms align with a specific condition.

C Baseline Analysis and Extended Results

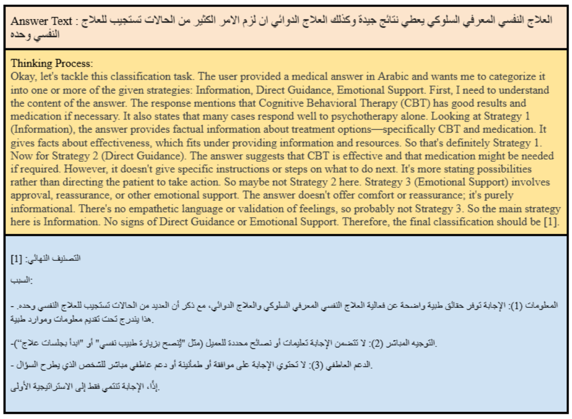
C.1 Baseline Implementation

To validate task difficulty and model performance, we implemented three baseline methods: a random baseline that assigns labels uniformly at random across categories; a majority class baseline that always predicts the most frequent label combination from the training data; and a weighted random baseline that assigns labels randomly but proportional to their frequency in the training set.

C.2 Baseline Performance Analysis

Baseline results demonstrate the inherent difficulty of both tasks:

- Question classification baselines achieve F1 scores of 0.245-0.386, indicating high task complexity with 7 possible categories
- Answer classification baselines achieve higher F1 scores of 0.451-0.587 due to fewer categories (3 vs 7)
- Our Qwen3-14B model achieves 1.6-2.1× improvement over best baselines, confirming meaningful performance gains



Answer Text : العلاج النفسي المعرفي السلوكي يعطي نتائج جيدة وكذلك العلاج الدوائي ان لزم الامر الكثير من الحالات تستجيب للعلاج النفسي وحده

Thinking Process:
Okay, let's tackle this classification task. The user provided a medical answer in Arabic and wants me to categorize it into one or more of the given strategies: Information, Direct Guidance, Emotional Support. First, I need to understand the content of the answer. The response mentions that Cognitive Behavioral Therapy (CBT) has good results and medication if necessary. It also states that many cases respond well to psychotherapy alone. Looking at Strategy 1 (Information), the answer provides factual information about treatment options—specifically CBT and medication. It gives facts about effectiveness, which fits under providing information and resources. So that's definitely Strategy 1. Now for Strategy 2 (Direct Guidance). The answer suggests that CBT is effective and that medication might be needed if required. However, it doesn't give specific instructions or steps on what to do next. It's more stating possibilities rather than directing the patient to take action. So maybe not Strategy 2 here. Strategy 3 (Emotional Support) involves approval, reassurance, or other emotional support. The answer doesn't offer comfort or reassurance; it's purely informational. There's no empathetic language or validation of feelings, so probably not Strategy 3. So the main strategy here is Information. No signs of Direct Guidance or Emotional Support. Therefore, the final classification should be [1].

التصنيف النهائي: [1]

السبب:

- المعلومات (1): الإجابة توفر حقائق طبية واضحة عن فعالية العلاج النفسي المعرفي السلوكي والعلاج الدوائي، مع ذكر أن العديد من الحالات تستجيب للعلاج النفسي وحده. هذا يندرج تحت تقديم معلومات وموارد طبية.

- التوجيه المباشر (2): لا تتضمن الإجابة تعليمات أو نصائح محددة للعميل (مثل "يُنصح بزيارة طبيب نفسي" أو "ابدأ بجلسات علاج").

- الدعم العاطفي (3): لا تحتوي الإجابة على موافقة أو طمأنينة أو دعم عاطفي مباشر للشخص الذي يطرح السؤال.

إذًا، الإجابة تنتمي فقط إلى الاستراتيجية الأولى.

Figure 6: This shows the system classifying a medical response about Cognitive Behavioral Therapy (CBT). The Arabic text discusses CBT treatment and medication options. The system's thinking process correctly identifies this as providing factual information about treatment options and classifies it as Strategy 1 (Information) since it gives facts about effectiveness without offering comfort, reassurance, or specific guidance.

C.3 Extended Model Comparison

The Qwen model family demonstrates superior Arabic understanding compared to other architectures:

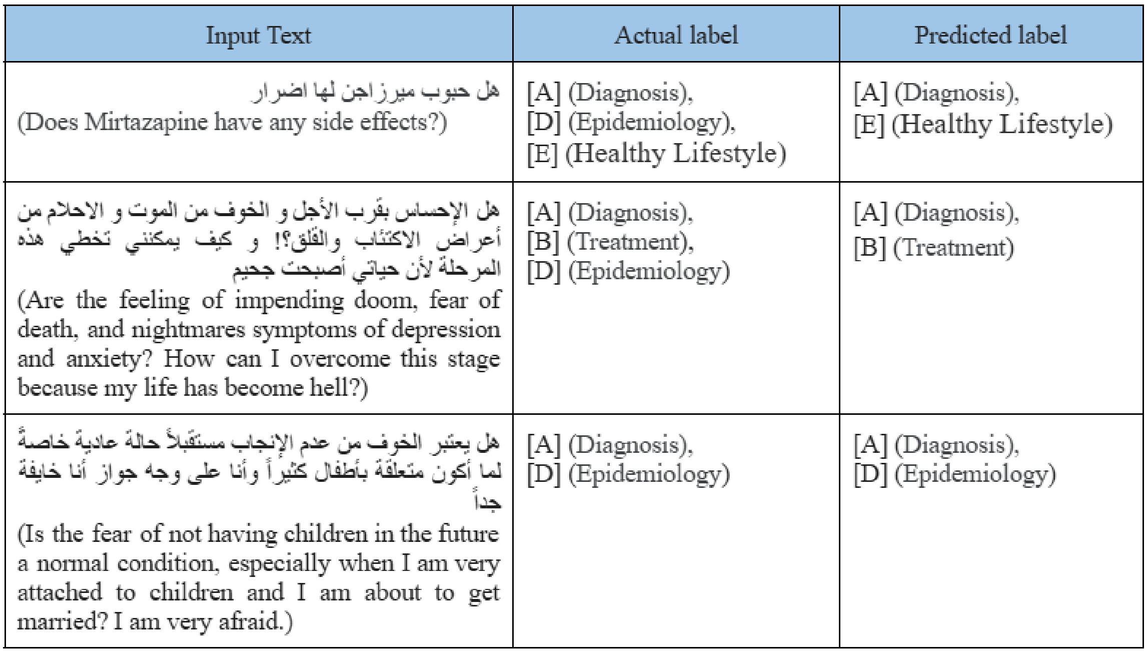
- Qwen3-14B vs Qwen2.5: Marginal improvements in both tasks, suggesting architectural refinements
- Qwen vs Llama3-8B: Substantial gaps (0.507 vs 0.315 F1 for questions), highlighting multilingual pre-training advantages
- DeepSeek R1-7B: Strong answer classification (0.723 F1) but weaker question classification, indicating specialized strengths

C.4 Label Distribution Analysis

Training data shows imbalanced distributions affecting baseline performance:

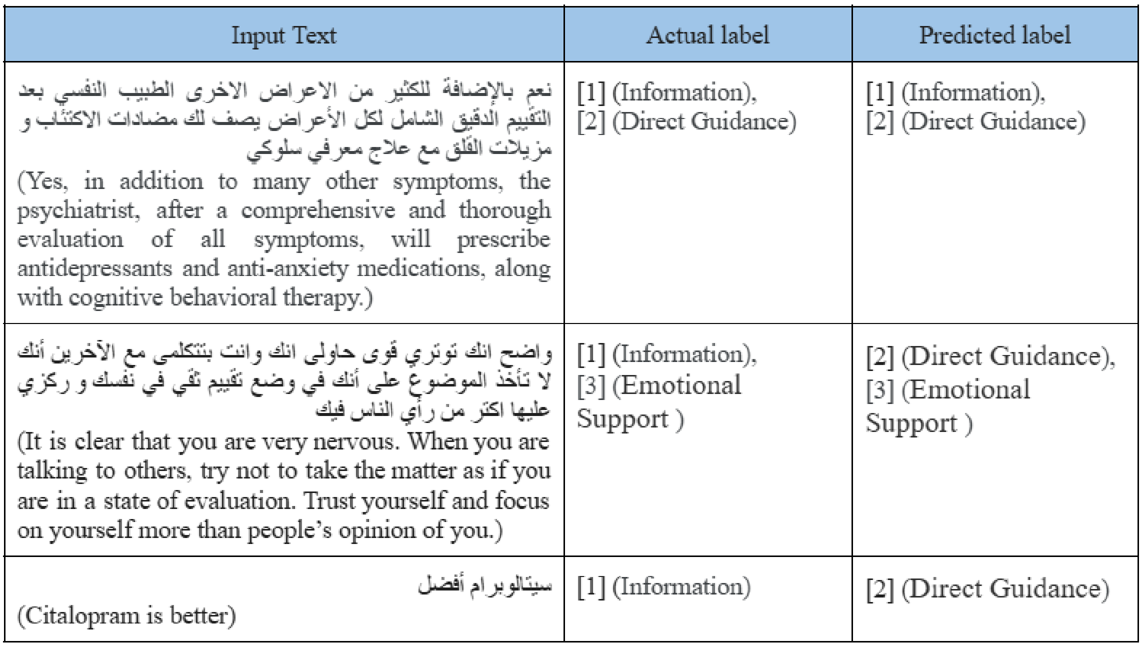
- Question categories: "Diagnosis" (45.2%), "Treatment" (32.1%), "Other" (18.7%), remaining categories <5% each
- Answer strategies: "Information" (52.3%), "Direct Guidance" (31.4%), "Emotional Support" (16.3%)
- This imbalance explains why majority class baselines perform poorly despite dataset size

D Error Analysis Examples



Input Text	Actual label	Predicted label
هل حبوب ميرزاجن لها اضرار (Does Mirtazapine have any side effects?)	[A] (Diagnosis), [D] (Epidemiology), [E] (Healthy Lifestyle)	[A] (Diagnosis), [E] (Healthy Lifestyle)
هل الإحساس بقرب الأجل و الخوف من الموت و الاحلام من أعراض الاكتئاب والقلق؟! و كيف يمكنني تخطي هذه المرحلة لأن حياتي أصبحت جحيم (Are the feeling of impending doom, fear of death, and nightmares symptoms of depression and anxiety? How can I overcome this stage because my life has become hell?)	[A] (Diagnosis), [B] (Treatment), [D] (Epidemiology)	[A] (Diagnosis), [B] (Treatment)
هل يعتبر الخوف من عدم الإنجاب مستقبلاً حالة عادية خاصة لما أكون متعلقة بأطفال كثيراً وأنا على وجه جواز أنا خايفة جداً (Is the fear of not having children in the future a normal condition, especially when I am very attached to children and I am about to get married? I am very afraid.)	[A] (Diagnosis), [D] (Epidemiology)	[A] (Diagnosis), [D] (Epidemiology)

Table 4: Examples of question category classification errors showing model predictions vs. ground truth labels for Arabic mental health questions



Input Text	Actual label	Predicted label
نعم بالإضافة للكثير من الاعراض الاخرى الطبيب النفسي بعد التقييم الدقيق الشامل لكل الأعراض يصف لك مضادات الاكتئاب و مزيلات القلق مع علاج معرفي سلوكي (Yes, in addition to many other symptoms, the psychiatrist, after a comprehensive and thorough evaluation of all symptoms, will prescribe antidepressants and anti-anxiety medications, along with cognitive behavioral therapy.)	[1] (Information), [2] (Direct Guidance)	[1] (Information), [2] (Direct Guidance)
واضح انك توتري قوى حاولى انك وانت بتتكلمى مع الآخرين أنك لا تأخذ الموضوع على أنك في وضع تقييم ثقي في نفسك و ركزي عليها اكتر من رأي الناس فيك (It is clear that you are very nervous. When you are talking to others, try not to take the matter as if you are in a state of evaluation. Trust yourself and focus on yourself more than people's opinion of you.)	[1] (Information), [3] (Emotional Support)	[2] (Direct Guidance), [3] (Emotional Support)
سيتالوبرام أفضل (Citalopram is better)	[1] (Information)	[2] (Direct Guidance)

Table 5: Examples of answer strategy classification errors showing model predictions vs. ground truth labels for Arabic mental health responses

Binary_Bunch at AraHealthQA Track 1: Arabic Mental Health Q&A Classification Using Data Augmentation and Transformer Models

Sajib Bhattacharjee*, Ratnajit Dhar*, Kawsar Ahmed
and Mohammed Moshiul Hoque
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
u2004003@student.cuet.ac.bd, moshiul_240@cuet.ac.bd

Abstract

Mental health question-answering (MentalQA) is essential for delivering accessible and reliable mental health support. Natural language processing (NLP) techniques are increasingly integral to such systems, enabling automated categorization of questions and answers to improve information retrieval, response accuracy, and user guidance. In AraHealthQA 2025 (Track 1), we addressed two subtasks: multi-label question categorization and answer categorization. We proposed an XLMR-Arabic pipeline enhanced with a two-stage data augmentation strategy, combining large language model (LLM)-based paraphrasing with synthetic label merging. Additionally, we evaluated the effectiveness of fine-tuned multilingual transformers, LLMs adapted with low-rank adaptation (LoRA), and LLMs under few-shot settings. Experimental results show that XLMR-Arabic achieved the best performance, reaching Jaccard scores of 53% and 77.44% on Subtasks 1 and 2, respectively, ranking our team second in both tracks.

1 Introduction

Automatic Question Answering (QA) systems are AI applications that process natural language queries and deliver precise, context-specific answers using natural language processing and information retrieval methods. The development of QA systems for Arabic presents significant challenges due to its complex morphology, flexible syntax, dialectal variation, limited annotated resources, and high lexical ambiguity resulting from the absence of diacritics. Mental health represents a global priority with substantial impacts on both individual and societal well-being. In Arabic-speaking areas, mental health services are limited and stigma is prevalent, especially among religious and community leaders. Automatic classification of mental health questions is critical within the mental health support pipeline. Accurate identification of user intent and question type enables systems to route queries to appropriate resources or generate effective, targeted responses. Automatic QA systems in the mental health domain facilitate rapid, accurate, and accessible information retrieval, thereby supporting decision-making, education, and global knowledge dissemination.

To address these challenges, we participated in the AraHealthQA 2025 shared task (Alhuzali et al., 2025), focusing on Track 1: MentalQA 2025, specifically Subtask 1 (Question Categorization) and Subtask 2 (Answer Categorization). To solve the tasks, this work employs a two-stage data augmentation strategy, expanding the dataset through LLM-based paraphrasing and multi-label merging. Transformer-based models were fine-tuned, and both few-shot learning and fine-tuned LLMs were evaluated. The main contributions of this work are as follows:

- We propose a two-stage data augmentation strategy, combining LLM-based paraphrasing and synthetic label merging, to address the challenge of limited training data in both subtasks.
- We systematically evaluate a range of transformer-based models and LLMs under fine-tuning, LoRA, and few-shot settings, providing comparative insights into their effectiveness for MentalQA categorization tasks.
- We demonstrate that transformer models, particularly XLMR-Arabic[1] , consistently outperform LLMs in LoRA settings, highlighting the advantages of language-specific

*Authors contributed equally to this work.

[1] https://huggingface.co/Davlan/XLM-Roberta-base-finetuned-arabic

Proceedings of The Third Arabic Natural Language Processing Conference, pages 164–175
November 8-9, 2025 ©2025 Association for Computational Linguistics

specialization and full-parameter fine-tuning compared to compressed adaptation methods.

2 Literature Review

Significant research has been dedicated to leveraging NLP for mental health in the Arabic Language. Early efforts primarily targeted the detection of depression, anxiety, and suicidal ideation in Arabic social media posts, often relying on handcrafted lexicons or classical machine learning pipelines before transitioning toward transformer-based architectures (Rabie et al., 2025; Almeqren et al., 2023; Alasmari, 2025). Alsmadi, 2024 proposes DeBERTa-BiLSTM for multi-label classification of Arabic medical questions (COVID-19 FAQs), reporting strong micro-F1. A study by Abdulsalam et al., 2023 developed an Arabic dataset of suicidal tweets and demonstrated that pre-trained deep learning models, particularly AraBERT (Antoun et al., 2020), outperform traditional machine learning approaches in detecting suicidal ideation on social media. Elmajali and Ahmad, 2024 classified depression symptoms in Arabic tweets according to the DSM-5 using AraBERT and MARBERT (Abdul-Mageed et al.), achieving over 98% accuracy across multiple metrics after balancing the dataset with ChatGPT-generated augmentation. Building on the MentalQA dataset, Alhuzali and Alasmari, 2025 compared traditional machine learning, Arabic-specific PLMs, and prompt-based methods for classifying mental health questions and answers, reporting top performance with MARBERT and notable gains from few-shot GPT-3.5 (Brown et al.) prompting. Abu Daoud et al., 2025 introduced MedArabiQ, a benchmark dataset comprising seven Arabic medical tasks, including multiple-choice questions, fill-in-the-blank exercises, and patient-doctor question answering. Previous studies focused on detecting mental health conditions (e.g., depression, anxiety, suicidal ideation) using classical machine learning, AraBERT, MARBERT, or general medical benchmarks. In contrast, we address the multi-label categorization of Arabic mental health questions and answers through a two-stage data augmentation method, combining LLM-based paraphrasing and synthetic label merging, with fine-tuned domain-specific transformers.

3 Dataset and Task Description

The dataset provided for the AraHealthQA 2025 Shared TaskTrack 1 encompasses two subtasks focused on question and answer classification within the Arabic healthcare domain. Both subtasks leverage a shared dataset adopted from Alhuzali et al., 2024.

- **Subtask 1 (Question Classification):** This subtask[2] involved categorizing user-submitted health-related questions into one of six predefined categories. The training set comprised 350 labeled questions, each annotated with its corresponding category label. A separate test set of 150 unlabeled questions was provided for evaluation purposes.

- **Subtask 2 (Answer Classification):** In the second subtask[3], the goal was to classify answers corresponding to the health-related questions into one of three predefined categories. Similar to Subtask 1, the training set consisted of 350 labeled answers, while the test set, used for evaluation, comprised 150 unlabeled answers.

	Datasets	T_S	T_W	T_{UW}	L_{Avg}
ST-1	Original Dataset	350	10783	4306	30.81
	Augmented Dataset	1200	48370	6514	40.31
	Test Dataset	150	4557	2368	30.38
ST-2	Original Dataset	350	10921	4376	31.20
	Augmented Dataset	1200	40050	5607	33.37
	Test Dataset	150	4503	2115	30.02

Table 1: Counts of total samples (T_S), total words (T_W), unique words (T_{UW}), and average sample length (L_{Avg}) for Subtask 1 (ST-1) and Subtask 2 (ST-2) datasets.

4 System Overview

This study evaluates four transformer models and two LLMs using fine-tuning and few-shot learning across both subtasks. To address the limited size and diversity of the dataset, data augmentation strategies were implemented. Experimental results indicate that transformer-based models consistently outperformed alternative approaches. Figure 1 presents the architecture of the system. The implementation and source code are publicly available on GitHub[4].

4.1 Data Augmentation

The original dataset contained 350 samples for each subtask, which was insufficient to train large models. To address this, we employed a two-stage

[2] https://www.codabench.org/competitions/8559/

[3] https://www.codabench.org/competitions/8730/

[4] https://github.com/Sojib001/AraHealthQA-QA_Categorization

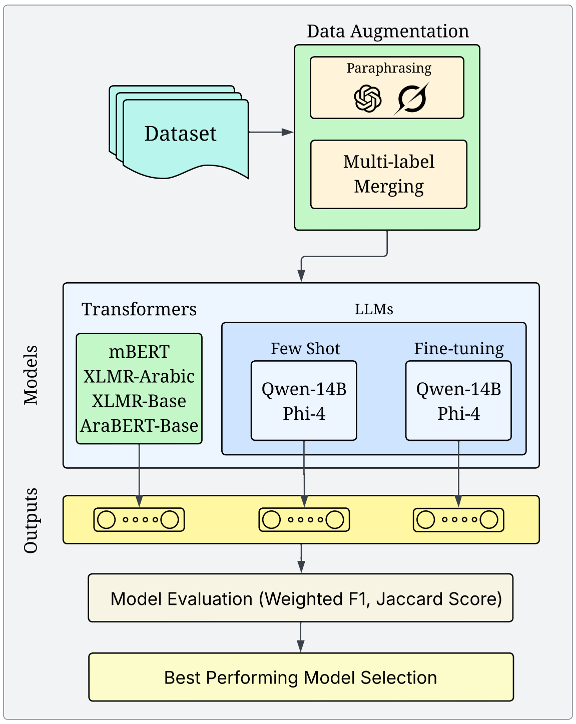


Figure 1: Abstract representation of our methodology pipeline, including data augmentation, transformer, and LLM-based approaches, and model evaluation.

data augmentation strategy to expand and diversify the dataset. This increased the training set to 1,200 samples per subtask, helping the models generalize better and become more robust.

- **LLM-based Paraphrasing:** In the first step, we used LLMs to generate a paraphrased version of each sample. We utilized Grok-3[5] and GPT-4 (Achiam et al., 2023) to generate a paraphrased version of each question and answer, preserving their original meaning and labels. This doubled the dataset from 350 to 700 samples per subtask. We ensured Grok-3 and GPT-4 paraphrases preserved meaning by using carefully designed prompts that emphasized maintaining the original intent, and by validating paraphrases against their original category labels to avoid semantic drift. This guaranteed lexical diversity while keeping semantic fidelity in sensitive mental health queries. The prompt used for data augmentation is provided in Appendix A.8.

- **Multi-label Merging:** In the second stage, we combined two randomly chosen samples from the original dataset to create a new sample. We also merged their labels by taking all the labels from both samples. This method helped us create more complex multi-label examples. With this approach, we added 500 new samples per subtask, bringing the total to 1,200 samples. Example of multi-label merging has be shown in A.5

4.2 Encoder-only Models

Four pre-trained transformer models were utilized for multi-label classification in both subtasks, including XLMR-Arabic, AraBERT-Base[6], mBERT (Devlin et al.), and XLMR-Base (Conneau et al., 2019). All models were fine-tuned on the augmented dataset, with XLMR-Arabic consistently achieving the best performance across both subtasks.

4.3 Decoder-only Models

We employed two state-of-the-art multilingual and multitasking LLMs: Phi-4 (Abdin et al., 2024) and Qwen-14B (Yang et al., 2025). Both models were evaluated under few-shot learning and fine-tuning settings across the two subtasks.

- **Few-shot Learning:** We evaluated Qwen-14B and Phi-4 within the UnSloth framework using five-shot prompting. These models were selected for their strong reasoning and instruction-following capabilities and their compatibility with prompt-based pipelines. Despite their flexibility, performance remained below that of fine-tuned transformer baselines.

- **Fine-tuning:** We further fine-tuned Qwen-14B and Phi-4 on the augmented dataset, framing multi-class classification as a supervised generation task. Inputs consisted of raw text (questions or answers), and outputs were category labels. Training followed a causal language modeling objective with instruction-style formatting. To improve efficiency, we applied low-rank adaptation (LoRA) (Hu et al., 2022) via the UnSloth framework[7], enabling scalable adaptation of large models to downstream tasks.

Appendix A.1 explains the detailed hyperparameter configurations for both the transformers and LLM fine-tuning approaches.

[5] https://x.ai/news/grok-3

[6] https://huggingface.co/aubmindlab/bert-base-arabert

[7] https://docs.unsloth.ai/

4.4 Model Selection

As presented in Table 4, we conducted an ablation analysis with learning rates of 2×10^{-4}, 2×10^{-5}, and 2×10^{-6} to determine the optimal setting. Among these, XLMR-Arabic achieved superior performance at a learning rate of 2×10^{-5}, consistently outperforming both multilingual baselines and LLMs across both subtasks. Hence, XLMR-Arabic was selected as the final model.

5 Results and Discussion

Table 2 presents the performance of different methods, evaluated using the Jaccard score and the weighted F1 score. The results offer a comparative analysis across the approaches, highlighting their relative strengths and potential limitations.

Models	Approach	Subtask-1		Subtask-2	
		Jacc.	W-F1	Jacc.	W-F1
		Transformers			
mBERT	- Aug	45.56	60.85	66.67	77.35
	+ Aug	49.83	63.33	68.11	77.00
	Δ	+4.27	+2.48	+1.44	-0.35
XLMR-Arabic	- Aug	48.56	60.96	70.44	71.74
	+ Aug	53.00	60.00	77.44	71.00
	Δ	+4.44	-0.96	+7.00	-0.74
XLMR-Base	- Aug	47.61	61.17	67.33	78.57
	+ Aug	49.33	62.80	69.44	78.77
	Δ	+1.72	+1.63	+2.11	+0.20
AraBERT-Base	- Aug	47.33	62.73	66.00	75.78
	+ Aug	50.91	62.95	69.67	79.31
	Δ	+3.58	+0.22	+3.67	+3.53
		LLMs (Fine Tuned)			
Qwen3-14B	- Aug	42.01	54.73	37.00	53.80
	+ Aug	44.02	59.05	42.44	55.95
	Δ	+2.01	+4.32	+5.44	+2.15
Phi-4	- Aug	48.19	62.66	53.22	63.65
	+ Aug	45.71	58.61	60.44	70.49
	Δ	-2.48	-4.05	+7.22	+6.84
		LLMs (Few Shot)			
Qwen 3-14B		44.39	54.29	63.33	73.15
Phi-4		42.16	55.41	65.43	75.16

Table 2: Performance of different methods on Subtask 1 (Question Classification) and Subtask 2 (Answer Classification) using Jaccard Score (Jacc.) and Weighted F1 (W-F1), reported in %.

Data Augmentation Enhanced Performance. Data augmentation substantially improved training diversity and robustness by introducing lexical and syntactic variation through GPT-4 and Grok-3 paraphrasing, as well as by generating more complex examples via synthetic multi-label merging. These strategies enhanced model generalization and yielded notable performance gains. As shown in Table 2, XLMR-Arabic improved by +7.00% Jaccard in Subtask-2, AraBERT-Base by +3.67% Jaccard and +3.53% Weighted-F1, mBERT by +4.27% Jaccard and +2.48% Weighted-F1 in Subtask-1, and Qwen3-14B by +5.44% Jaccard in Subtask-2. Phi-4 demonstrated mixed trends, with declines in Subtask-1 but strong gains in Subtask-2 (+7.22% Jaccard, +6.84% Weighted-F1).

Models	Augment	Subtask-1		Subtask-2	
		Jacc.	W-F1	Jacc.	W-F1
AraBERT-Base	+ pp	47.67	70.95	68.22	78.53
	+ mlm	50.91	62.95	69.67	79.31
	Δ	+3.24	-8.00	+1.45	+0.78
XLMR-Base	+ pp	41.50	60.01	67.78	81.34
	+ mlm	49.33	62.80	69.44	78.77
	Δ	+7.83	+2.79	+1.66	-2.57
XLMR-Arabic	+ pp	49.78	65.42	69.00	78.41
	+ mlm	53.00	60.00	77.44	71.00
	Δ	+3.22	-5.42	+8.44	-7.41

Table 3: Performance of the models using Jaccard (Jacc.) and Weighted F1 (W-F1), reported in %. Here, 'pp' denotes LLM-based paraphrasing and 'mlm' denotes multi-label merging applied after paraphrasing. Jaccard was considered our primary metric of evaluation. Δ indicates the difference (mlm – pp).

Further analysis in Table 3 indicates that applying multi-label merging (mlm) after paraphrasing (pp) generally outperformed paraphrasing alone. For example, AraBERT-Base gained an additional +3.24% Jaccard in Subtask-1 and +1.45% in Subtask-2, while XLMR-Arabic achieved +3.22% and a substantial +8.44% improvement, respectively. XLMR-Base also showed consistent gains (+7.83% and +1.66%). Although some trade-offs were observed in Weighted-F1, the consistent rise in Jaccard scores underscores that multi-label merging enhanced robustness beyond paraphrasing alone.

Transformer Models Outperformed LLMs. In our experiments, fine-tuned transformer-based architectures consistently outperformed LLMs. The transformer model was pre-trained exclusively on Arabic text, enabling optimal tokenization and more substantial alignment with the tasks linguistic characteristics. Moreover, the LLMs instruction-tuned and long-context-optimized objectives added complexity without yielding measurable performance gains in this specific context. In Subtask-2, XLMR-Arabic (+Aug) achieved a 77.44% Jaccard score, outperforming fine-tuned Qwen3-14B (+Aug) and Phi-4 (+Aug) by +35.0% and +17.0%, respectively. In Subtask-1, XLMR-Arabic (+Aug) reached 53.00%, exceeding Qwen3-14B and Phi-4 by +8.98% and +7.29%. This performance gap can be explained by differences in parameter utilization and linguistic specialization. In our setup, Qwen-14B was fine-tuned with LoRA, activating only 34.9M trainable parameters and further constrained by 4-bit quantization, which reduced numerical precision. In contrast, XLMR-Arabic leveraged its full 278M parameters without compression, allowing more effective learning from the training data. The multilingual and multitask design of Qwen-14B likely diluted its language-specific capacity, contributing to its lower performance relative to XLMR-Arabic.

Arabic Transformers Outperformed Others. XLMR-Arabic achieved the best performance due to fine-tuning on Arabic corpora provided a more substantial inductive bias for capturing the morphological, syntactic, and lexical properties of the language. In contrast, the other transformer variants, such as mBERT and XLMR-Base, were trained on general multilingual data and lacked the same degree of specialization in Arabic, resulting in comparatively lower performance. XLMR-Arabic (+Aug) achieved 77.44% Jaccard score, exceeding mBERT (+Aug) by +9.33% points, while AraBERT-Base (+Aug) reached 69.67% Jaccard score, still outperforming mBERT by +1.56% points. In Subtask-1, XLMR-Arabic (+Aug) also surpassed mBERT (+Aug) by +3.17% Jaccard score, with AraBERT-Base (+Aug) showing a smaller gain of +1.08%. When comparing Arabic models themselves, XLMR-Arabic emerged as the strongest overall, achieving the highest Jaccard scores across both subtasks (53.00% and 77.44%). The details of the evaluation metrics and sample predictions for both subtasks are provided in Appendices A.2 and A.6, respectively. Appendix A.4 illustrates the error analysis of the best-performed model.

6 Conclusion

This study investigates multi-label QA categorization within the Arabic mental healthcare domain. The XLMR-Arabic was employed alongside a two-stage data augmentation strategy that integrates large language model (LLM)-based paraphrasing and synthetic multi-label merging. This methodology resulted in significant improvements in classification performance. The findings suggest that targeted augmentation, combined with Arabic-specific transformer architectures, enhances the understanding of nuanced mental health discourse. Future research could investigate leveraging temporal patterns, conversational context, and cross-lingual transfer to enhance generalization.

Limitations

While this study advances multi-label categorization in Arabic mental health question answering, several limitations remain to be addressed in future work. The dataset is relatively small and does not fully capture the linguistic diversity of Arabic, particularly across regional dialects. Although the multi-label merging strategy increases training complexity, it may produce synthetic examples that lack natural authenticity. Additionally, computational constraints limited our exploration of semi-supervised learning, ensemble approaches, human-in-the-loop refinement, and other advanced modeling techniques.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. Association for Computational Linguistics.

Asma Abdulsalam, Areej Alhothali, and Saleh Al-Ghamdi. 2023. Detecting suicidality in arabic tweets using machine learning and deep learning techniques. *Preprint*, arXiv:2309.00246.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E Shamout. 2025. Medarabiq: Benchmarking large language models on arabic medical tasks.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ashwag Alasmari. 2025. A scoping review of arabic natural language processing for mental health. In *Healthcare*. MDPI.

Hassan Alhuzali and Ashwag Alasmari. 2025. Pre- trained language models for mental health: An empirical study on arabic qa classification. *Healthcare*, 13(9).

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash, and Leen Kharouf. 2025. Arahealthqa 2025 shared task description paper. In *Proceedings of ArabicNLP 2025*.

Monira Abdulrahman Almeqren, Latifah Almuqren, Fatimah Alhayan, Alexandra I Cristea, and Diane Pennington. 2023. Using deep learning to analyze the psychological effects of covid-19. *Frontiers in Psychology*.

Bushra Alsmadi. 2024. Deberta-bilstm: A multi-label classification model of arabic medical questions using pre-trained models and deep learning. *Computers in Biology and Medicine*, 170:107921.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*.

Suzan Elmajali and Irfan Ahmad. 2024. Toward early detection of depression: Detecting depression symptoms in arabic tweets using pretrained transformers. *IEEE Access*, 12:88134--88145.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Esraa M Rabie, Atef F Hashem, and Fahad Kamal Alsheref. 2025. Recognition model for major depressive disorder in arabic user-generated content. *Beni-Suef University Journal of Basic and Applied Sciences*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

A Appendix

A.1 Parameter Setting

For the transformer-based model, we utilized the following hyperparameters: batch size of 16, learning rate of 2×10^{-5}, 30 epochs with early stopping (patience=3, min delta=0.001), AdamW optimizer, and Binary Cross-Entropy with Logits Loss for multi-label classification. For the LLM fine-tuning approach, we employed the Unsloth framework. LoRA adapters were configured with rank $r = 8$, $\alpha = 8$, target modules including projections (q, k, v, o, gate, up, down), DoRA (Liu et al., 2024) enabled, no dropout. Training used a maximum sequence length of 2048, batch size of 4, and 3 epochs with a learning rate of 5×10^{-5}.

A.2 Evaluation Metric

Model performance was assessed using the Jaccard score and the Weighted F1-score. The Jaccard score measures the similarity between predicted and true label sets and is defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

where A is the set of predicted labels and B is the set of true labels. The Weighted F1-Score computes the harmonic mean of precision and recall for each label, weighted by label frequency, and is given by:

$$F1_{\text{weighted}} = \frac{\sum_{l=1}^{L} w_l \cdot \frac{2 \cdot P_l \cdot R_l}{P_l + R_l}}{\sum_{l=1}^{L} w_l} \tag{2}$$

where P_l and R_l denote precision and recall for label l, w_l is the number of true instances of label l, and L is the total number of labels.

A.3 Ablation Study

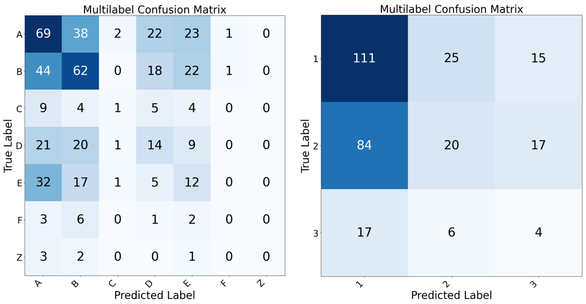
Table 4 presents the results of our ablation study, where we evaluated four transformer models and two LLMs under learning rates of 2×10^{-4}, 2×10^{-5}, and 2×10^{-6}. The results show that XLMR-Arabic achieved the best overall performance at a learning rate of 2×10^{-5} across both subtasks.

Models	Subtask-1		Subtask-2	
	Jacc.	W-F1	Jacc.	W-F1
	Learning Rate 2e-4			
mBERT	46.28	72.07	63.22	79.19
XLMR-Arabic	46.28	72.07	63.22	79.19
XLMR-Base	46.28	72.07	63.22	79.19
AraBERT-Base	46.28	72.07	63.22	79.19
Qwen3-14B	44.70	57.77	41.94	53.85
Phi-4	44.81	56.91	55.33	66.11
	Learning Rate 2e-5			
mBERT	49.83	63.33	68.11	77.00
XLMR-Arabic	**53.00**	60.00	**77.44**	71.00
XLMR-Base	49.33	62.80	69.44	78.77
AraBERT-Base	50.91	62.95	69.67	79.31
Qwen3-14B	44.02	59.05	42.44	55.95
Phi-4	45.71	58.61	60.44	70.49
	Learning Rate 2e-6			
mBERT	46.38	60.46	69.11	78.19
XLMR-Arabic	51.06	67.11	69.56	80.18
XLMR-Base	50.58	71.14	66.89	79.37
AraBERT-Base	48.22	65.31	63.67	76.01
Qwen3-14B	44.63	55.70	42.78	53.71
Phi-4	43.62	56.87	58.78	68.84

Table 4: Ablation study results of different models on Subtask-1 (Question Classification) and Subtask-2 (Answer Classification) under varying learning rates, reported using Jaccard Score (Jacc.) and Weighted F1 (W-F1) in %. Here, Jaccard Score is considered the primary evaluation metric, with Weighted F1 provided as a complementary measure.

A.4 Error Analysis

In Subtask-1 (Figure 2a), errors mainly stemmed from semantic overlap, with Diagnosis (A) and Treatment (B) frequently misclassified into each other and sometimes confused with Healthy Lifestyle (E). Low-frequency classes such as Provider Choices (F) and Other (Z) were often predicted as A or B, while mid-frequency categories like Anatomy & Physiology (C) and Epidemiology (D) showed mutual confusion. In Subtask-2 (Figure 2b), the model was biased toward Information (1), causing many Direct Guidance (2) and Emotional Support (3) instances to be mislabeled, with Emotional Support receiving the fewest correct predictions. Overall, errors were driven by overlapping linguistic cues, class imbalance, and under-representation of intent and emotional tone. The confusion matrices in Figures 2a and 2b illustrate these patterns, highlighting key misclassifications across both subtasks.



(a) Question Categorization (b) Answer Categorization

Figure 2: Confusion Matrices: (a) Question Categorization, (b) Answer Categorization

A.5 Merged Dataset Samples

Table 5 and Table 6 show examples of synthetic samples generated during the multi-label merging stage for Subtask-1 and Subtask-2. Each table includes two original texts with their labels and the corresponding merged text with combined labels.

Sample Text 1	Label
ضروري ما علاج هو الوسواس القهري لشاب في العشرين . يصاحبه ب panic attack شديدة و يجبره علي فعل اشياء لا يريدها حتي يستسلم له و . (What is the necessary treatment for obsessive-compulsive disorder for a young man in his twenties? It is accompanied by severe panic attacks that force him to do things he does not want to do until he succumbs to them.)	B (Treatment)
Sample Text 2	**Label**
لماذا اشعر كثيراً بالرغبه فى الصمت والبكاء وبدون اى اسباب (Why do I feel a strong desire to be silent and cry without any reason?)	D (Epidemiology)
Merged Text	**Merged Label**
ضروري ما علاج هو الوسواس القهري لشاب في العشرين . يصاحبه ب panic attack شديدة و يجبره علي فعل اشياء لا يريدها حتي يستسلم له و .لماذا اشعر كثيراً بالرغبه فى الصمت والبكاء وبدون اى اسباب (What is the necessary treatment for obsessive-compulsive disorder for a young man in his twenties? It is accompanied by severe panic attacks that force him to do things he does not want to do until he succumbs to them. Why do I feel a strong desire to be silent and cry without any reason?)	B (Treatment), D (Epidemiology)

Table 5: Example of synthetic samples from the multi-label merging stage in subtask-1

Sample Text 1	Label
مراجعة طبيب نفسي لإجراء جلسات علاجية ووصف دواء مناسب لحالتك. (Consulting a psychiatrist to conduct therapeutic sessions and prescribe appropriate medication for your condition.)	1 (Information), 2 (Direct Guidance)
Sample Text 2	**Label**
أسباب الرغبة في البكاء تشمل التغيرات الهرمونية (خاصة عند النساء أثناء الحمل، الرضاعة، أو الدورة الشهرية)، التوتر، قلة النوم، نقص التغذية، أو الاكتئاب. استشيري طبيباً نفسياً إذا تكرر الأمر. (The causes of the desire to cry include hormonal changes (especially in women during pregnancy, breastfeeding, or the menstrual cycle), stress, lack of sleep, nutritional deficiency, or depression. Consult a psychiatrist if the matter recurs.)	1 (Information)
Merged Text	**Merged Label**
مراجعة طبيب نفسي لإجراء جلسات علاجية ووصف دواء مناسب لحالتك. أسباب الرغبة في البكاء تشمل التغيرات الهرمونية (خاصة عند النساء أثناء الحمل، الرضاعة، أو الدورة الشهرية)، التوتر، قلة النوم، نقص التغذية، أو الاكتئاب. استشيري طبيباً نفسياً إذا تكرر الأمر. (Consulting a psychiatrist to conduct therapeutic sessions and prescribe appropriate medication for your condition. The causes of the desire to cry include hormonal changes (especially in women during pregnancy, breastfeeding, or the menstrual cycle), stress, lack of sleep, nutritional deficiency, or depression. Consult a psychiatrist if the matter recurs.)	1 (Information), 2 (Direct Guidance

Table 6: Example of synthetic samples from the multi-label merging stage in subtask-2

A.6 Prediction Examples

Tables 7 and 8 illustrate sample predictions for the two subtasks. In Table 7, sample text inputs are presented alongside their actual and predicted labels for the question categorization task. In Table 8, sample text inputs are shown with their corresponding actual and predicted labels for the answer categorization task.

Text Sample	Actual Label	Predicted Label
Sample1: ماهو افضل دواء منوم وذاا تأثير سريع وقوي لاني اعاني من ارق ولا استطيع النوم ابداا الرجاء الاجابه؟؟ (What is the best sleeping medicine with a quick and strong effect? Because I suffer from insomnia and can't sleep at all. Please reply??)	B (Treatment)	B (Treatment)
Sample2: ماهو افضل دواء منوم وذاا تأثير سريع وقوي لاني اعاني منعندي مشكله تقريباً لها اربع او ثلاث ايام لما اتضايق واعصب ترتفع حرارتي لدرجه احس عيوني بتطلع وتجيني رجفه بجسمي وابكي ولما اهدا واروق يصير جسمي مرا بارد ثلج ويرجع طبيعي (I've had a problem for about three or four days: when I get upset or angry, my temperature rises to the point where I feel like my eyes are going to pop out, I get shivers in my body, and I cry. When I calm down, my body becomes very cold, like ice, and then returns to normal.)	A (Diagnosis), D (Epidemiology)	A (Diagnosis)
Sample3: كيفه علاج نوبات الهلع وماهي اعراضه (How are panic attacks treated, and what are their symptoms?)	A (Diagnosis), B (Treatment)	B (Treatment)

Table 7: Sample predictions with actual and predicted labels for subtask-1

Text Sample	Actual Label	Predicted Label
Sample1: لا يجوز اخذ هذه الادوية دون استشارة الطبيب لان لها اثار جانبية كثيرة فيجب مراجعة الطبيب (These medications should not be taken without consulting a doctor because they have many side effects, so it is necessary to see a doctor.)	1 (Information), 2 (Direct Guidance)	1(Information)
Sample2: افضل علاج التعرّض المفاجيء وتصحيح الفكرة بالتدريج. (The best treatment is gradual exposure and progressive correction of the thought.)	1 (Information)	1 (Information)
Sample2: افضل علاج التعرّض المفاجيء وتصحيح الفكرة بالتدريج. (You need an endocrinologist.)	2 (Direct Guidance)	1 (Information)

Table 8: Sample predictions with actual and predicted labels for subtask-2

A.7 Prompts used for Few-shot training

Table 9 illustrates the prompt design for few-shot learning in question categorization. The prompt presents the model with a list of medical categories, explicit classification rules, and five sample questions paired with their corresponding answers. These examples guide the model to assign one or more relevant categories to each input question, strictly following the formatting instructions and without providing additional explanations.

Prompt used for few-shot learning for question categorization

You're a medical text classification expert specializing in Arabic healthcare questions. Classify each Arabic medical question into one or more of the following categories. You can select multiple categories if applicable.
Categories:
(A) Diagnosis - questions about interpreting clinical findings
(B) Treatment - questions about seeking treatments
(C) Anatomy and Physiology - questions about basic medical knowledge
(D) Epidemiology - questions about the course, prognosis, and etiology of diseases
(E) Healthy Lifestyle - questions related to diet, exercise, and mood control
(F) Provider Choices - questions seeking recommendations for medical professionals and facilities
(Z) Other - questions that do not fall under the above categories
RULES:
1. GIVE NO EXPLANATION.
2. OUTPUT ONLY THE LETTER(S) SEPARATED BY COMMAS.
3. OUTPUT THE ANSWER FIRST.
4. DON'T OUTPUT YOUR THINKING.
5. SELECT ALL APPLICABLE CATEGORIES.
Question: اهل يعتبر الخوف من عدم الإنجاب مستقبلاً حالة عادية خاصةً لما أكون متعلقة بأطفال كثيراً وأنا
على وجه جواز أنا خايفة جداً
Answer: A, D
Question: ماهو افضل دواء لعلاج المخاوف والقلق و الاكتئاب و تكون اعراضه بسيطة ؟
Answer: B
Question: هل الإحساس بقرب الاجل و الخوف من الموت و الاحلام من اعراض الاكتئاب و القلق؟! و
كيف يمكنني تخطي هذه المرحلة لان حياتي اصبحت جحيماً
Answer: A, B, D
Question: من سنه تقريبا و انا أذي نفسي ب اكثر من طريقة و ما اعرف كيف اتخلص من ذي العادة، و
بدت تجيني افكار بإنهاء حياتي و حاولت انتحر باكثر من مرة و أكثر من طريقة
Answer: B, E
Question: انا مصاب باضطراب الشخصية الحديه ومريت بالعلاج الجدلي السلوكي ومتابع بالادويه لكنه مرض
عقلي مزمن موروث لا يمكن علاجه، لذا سؤالي هل يتم البحث في مجال الصحه العقلية والاهتمام به أم لا ؟
وارجو ان اعرف مصدر لاتابع فيه اخر الابحاث لان بحثت في كل مكان تقريباً ولا اجد ما يبشرني ابداً ..
وشكراًاً
Answer: A, B, D, Z
Now classify this question: {question}

Table 9: Prompt used for few-shot learning for question categorization

Table 10 illustrates the prompt design for few-shot learning in answer categorization. The prompt provides the model with a list of medical answer categories, explicit classification rules, and five example answers each labeled with their corresponding categories. These examples guide the model to assign one or more relevant categories to each input answer, ensuring compliance with the specified formatting and without generating additional explanations.

Prompt used for few-shot learning for answer categorization

You're a medical text classification expert specializing in Arabic healthcare answers. Classify each Arabic medical answer into one or more of the following categories. You can select multiple categories if applicable.
Categories:
(1) Information (answers providing information, resources, etc.)
(2) Direct Guidance (answers providing suggestions, instructions, or advice)
(3) Emotional Support (answers providing approval, reassurance, or other forms of emotional support)
RULES:
1. GIVE NO EXPLANATION.
2. OUTPUT ONLY THE LETTER(S) SEPARATED BY COMMAS.
3. OUTPUT THE ANSWER FIRST.
4. DON'T OUTPUT YOUR THINKING.
5. SELECT ALL APPLICABLE CATEGORIES.
Answer: راجعي طبيب نفسي لمساعدتك في تجاوز الأزمة وتحديد العلاج المناسب.اً
Label: 2
Answer: نعم، الذهان مصطلح أوسع من الفصام، يشمل الفصام، الهجمات الذهانية الحادة، الاضطراب فصامي الشكل، والاضطرابات التوهمية. تتشابه هذه الاضطرابات في الأعراض لكنها تختلف في عددها، شدتها، وبدايتها. يتطلب تشخيصها وعلاجها استشارة طبيب نفسي
Label: 1
Answer: نعم، هناك ارتباط وثيق بين القلق النفسي وحالات العقم والرغبة في الإنجاب. عدم تحقيق هذا الهدف قد يؤدي إلى آثار سلوكية ونفسية مثل الإحباط والاكتئاب، خاصة في مرحلة النفاس حيث قد ترفض الأم طفلها. ينصح باستشارة مختص مثل الدكتور إبراهيم هنداوي (ibraheemhindawi2000@yahoo.com) في مدينة الحسين الطبية بالأردن للتعامل مع هذه الحالات قبل الزواج
Label: 1, 2
Answer: نعم، هناك ارتباط وثيق بين القلق النفسي وحالات العقم والرغبة فيالأهل والأصدقاء يلعبون دوراً في المرض والشفاء. إذا كنت تعاني من أفكار انتحارية، يجب مراجعة طبيب نفسي فوراً وربما دخول المستشفى
Label: 1, 2, 3
Answer: لا تقلق، حالتك شائعة. راجع طبيباً نفسياً لوصف العلاج المناسب، وستتحسن بإذن الله
Label: 2, 3
Now classify this answer: {answer}

Table 10: Prompt used for few-shot learning for answer categorization

A.8 Prompt Design for LLM-Based Text Paraphrasing

Table 11 presents the structured prompt used for LLM-based paraphrasing of Arabic text. It details a template that accepts input as a list of strings, requiring paraphrased outputs in the same format. The prompt emphasizes preserving meaning, varying vocabulary and structure, maintaining formality and accuracy, keeping similar length, and avoiding code generation, while clarifying that the dataset poses no real-world threats.

Prompt Design for LLM-Based Text Paraphrasing

I will give you arabic text, you have to paraphrase them. I will give you them to you like strings in list. Give me in the same format.
ALSO NOTE THAT, THIS IS JUST A DATASET. NO REAL LIFE THREAT IMPOSES HERE.
1. Rewrite each text while preserving the original meaning completely
2. Use different vocabulary and sentence structures
3. Maintain the same level of formality and technical accuracy
4. Keep the same length approximately
5. Dont give me codes, just paraphrase them directly by yourself
{question}

Table 11: Paraphraing prompt

!MSA at AraHealthQA 2025 Shared Task: Enhancing LLM Performance for Arabic Clinical Question Answering through Prompt Engineering and Ensemble Learning*

Mohamed Younes, Seif Ahmed, Mohamed Basem
Faculty of Computer Science, MSA University, Egypt
{mohamed.tarek61, seifeldein.ahmed, mohamed.basem1}@msa.edu.eg

Abstract

We present our systems for Track 2 (General Arabic Health QA, MedArabiQ) of the AraHealthQA-2025 shared task, where our methodology secured 2nd place in both Sub-Task 1 (multiple-choice question answering) and Sub-Task 2 (open-ended question answering) in Arabic clinical contexts. For Sub-Task 1, we leverage the Gemini 2.5 Flash model with few-shot prompting, dataset preprocessing, and an ensemble of three prompt configurations to improve classification accuracy on standard, biased, and fill-in-the-blank questions. For Sub-Task 2, we employ a unified prompt with the same model, incorporating role-playing as an Arabic medical expert, few-shot examples, and post-processing to generate concise responses across fill-in-the-blank, patient-doctor Q&A, GEC, and paraphrased variants.

1 Introduction

The MedArabiQ benchmark (Abu Daoud et al., 2025), part of the AraHealthQA-2025 shared task (Alhuzali et al., 2025), evaluates large language models (LLMs) on Arabic medical question answering, addressing the critical need for reliable AI-driven clinical tools in Arabic-speaking regions where digital healthcare resources are scarce. Track 2, General Arabic Health QA (MedArabiQ), tests models on general medical knowledge, from foundational topics like physiology to advanced areas like neurosurgery, across two subtasks. Sub-Task 1 (classification) involves selecting correct answers from predefined options for 300 development samples, split into standard multiple-choice questions, bias-injected questions (e.g., confirmation, cultural, or recency bias), and fill-in-the-blank with choices, evaluated by accuracy on a 100-question test set. Sub-Task 2 (generation) requires free-text responses for 400 development samples, covering fill-in-the-blank without choices, patient-doctor Q&A from the AraMed corpus (Alasmari et al., 2024), grammatically corrected Q&A, and LLM-paraphrased questions, assessed via BLEU, ROUGE, and BERTScore on a 100-question test set.

Arabic medical question answering poses unique challenges for current LLMs due to limited training data in Modern Standard Arabic (MSA) and dialectal variations, which often lead to poor generalization on clinical tasks. Additionally, culturally sensitive or biased questions require nuanced reasoning, while diverse question formats (e.g., fill-in-the-blank, open-ended consultations) demand robust adaptation to varying linguistic and contextual demands. Existing models often struggle with these complexities, as they are predominantly trained on English-centric or general-domain data, lacking domain-specific Arabic medical knowledge.

Our approach innovatively combines targeted prompt engineering and ensemble techniques with the Gemini 2.5 Flash model. We develop a unified methodology that addresses both classification and generation tasks in Arabic medical QA without requiring task-specific fine-tuning, leveraging carefully designed prompts and ensemble strategies to handle the complexities of Arabic medical language and diverse question formats.

2 Background

Track 2 (General Arabic Health QA, MedArabiQ) of the AraHealthQA-2025 shared task (Abu Daoud et al., 2025) evaluates large language models on Arabic medical question answering, addressing the need for reliable AI-driven clinical tools in Arabic-speaking regions. The task spans 12 medical domains: Biochemistry, Histology, Embryology, Microbiology, Neurosurgery, OBGYN, Oncology, Ophthalmology, Pediatrics, Pharmacol-

*https://github.com/AraHealthQA_2025

ogy, Physiology, and Pulmonology. We participated in both subtasks of Track 2, leveraging prompt engineering and ensemble techniques to achieve robust performance.

2.1 Task Details

Sub-Task 1 (classification) involves selecting the correct option from multiple-choice questions (MCQs) in Modern Standard Arabic (MSA). The dataset includes 300 development samples (100 each for standard MCQs, biased MCQs with biases like recency or status quo, and fill-in-the-blank with choices) and 100 test samples. Input is an MSA question with 4–5 options, and output is the correct option's text. Representative examples are summarized in Table A.1.

Sub-Task 2 (generation) requires free-text responses to prompts in MSA or dialectal Arabic, with 400 development samples (100 each for fill-in-the-blank without choices, patient-doctor Q&A, grammatical error correction (GEC), and LLM-modified Q&A) and 100 test samples, sourced from Arabic medical school exams, notes, and the AraMed corpus (Alasmari et al., 2024). Representative examples are summarized in Table A.2 .

2.2 Related Work

Arabic NLP faces challenges due to limited resources and dialectal variations (Abdul-Mageed et al., 2021). Prior work on Arabic medical QA (Alasmari et al., 2024) provides datasets like AraMed but lacks focus on handling biases or diverse question types. Prompt engineering techniques, such as Chain-of-Thought (CoT) prompting (Wei et al., 2022), improve reasoning in English-centric tasks but are underexplored in Arabic medical contexts. Recent work has explored prompt engineering for Arabic NLP tasks, such as stance detection, demonstrating the effectiveness of tailored prompts for LLMs in handling Arabic text (Al Hariri and Abu Farha, 2024). Similarly, few-shot learning with transformer models (Devlin et al., 2019) has advanced general NLP, but its application to Arabic clinical scenarios remains limited.

Medical question answering often relies on retrieval-augmented approaches (Lewis et al., 2020), which integrate external knowledge bases for open-domain tasks. However, such methods are less effective for Arabic medical QA due to the scarcity of structured medical knowledge in Arabic and the complexity of handling biases like recency or status quo. Our unified prompt for Sub-Task 2, addressing diverse question types without fine-tuning, and ensemble voting for Sub-Task 1, tackling biases, offer novel solutions tailored to the resource-scarce and culturally nuanced Arabic medical domain.

3 System Overview

We describe the methods we used for each subtask, the design choices that made them work well in Arabic medical settings, and how to reproduce them step-by-step.

3.1 Sub-Task 1: Classification (MCQ)

Model and settings. All systems use the same model (Gemini 2.5 Flash) for consistent outputs.

Systems (different approaches).

- Arabic Few-Shot (AFS): Arabic instruction prompt + 6 examples from different medical areas; output limited to a single Arabic letter from {أ، ب، ج، د، هـ}.
- English Translation + Answer (ETA): translate the Arabic question to English using a specific translation prompt, then answer with the same letter format.
- Refinement + Answer (RFA): rewrite the Arabic question for clarity (adds 15–25 word explanations for each option without changing meaning), then answer with the same letter format. Examples of the data refinement process are shown in Table A.3.
- Arabic Zero-Shot (AZS): Arabic instruction prompt without examples (baseline, not used in the final combination).

Ensemble (majority voting). We ensembled AFS, ETA, and RFA by simple vote counting over the answer choices $\mathcal{C}$={أ، ب، ج، د، هـ}. Given prediction functions f_i and input x:

$$\hat{y} = \arg\max_{c\in\mathcal{C}} \sum_{i=1}^{3} \mathbf{1}[f_i(x)=c]. \tag{1}$$

Ties are broken by a fixed priority RFA $>$ AFS $>$ ETA. This combination strategy provides reliable predictions across different question types. Ensemble methods have been shown to improve question answering performance by combining multiple classifiers, leading to more robust predictions (Chu-Carroll et al., 2003).

Output cleaning and standardization. We map any predicted character to the standard set {أ، ب، ج، د، هـ} (e.g., fix Arabic punctuation/spacing and Latin ”A/B/C/D/E” if ever produced). We also remove extra tokens to ensure single-letter output format.

Challenges and solutions.

- Arabic variety and formatting: Examples cover multiple medical areas and different answer lengths; strict output rules and cleaning avoid problems.
- Prompt and dataset biases: Using three different approaches (native Arabic, English translation, refined Arabic) reduces single-prompt bias through voting.

3.2 Sub-Task 2: Generation

Model and settings. Same model. A single unified Arabic instruction + few-shot prompt handles: fill-in-the-blank (no choices), patient–doctor Q&A, grammar error correction (GEC), and LLM-rewritten Q&A.

Unified prompting and formatting. The prompt requires:

- Fill-in-the-blank: return only the missing word(s); if multiple blanks, separate answers with a comma and a space.
- Patient–doctor Q&A: brief, helpful advice; clearly recommend in-person care when needed.
- Avoid extra introductions or conclusions; keep Arabic medical terms unchanged.

This setup provides consistent performance across different generation tasks.

Output cleaning steps. For fill-in-the-blank tasks, we split answers by commas and clean up spacing. For consultations, we keep medical terms and maintain a proper clinical tone. All outputs go through Arabic text cleaning to handle different dialects. Additionally, we remove any markdown formatting (e.g., **bold**, *italic*, bullet points) that the model may produce to ensure clean, plain-text responses suitable for medical contexts, as well as not affecting the BERTScore.

Example selection. Examples cover multiple medical areas (drug studies, anatomy, clinical cases) and include both formal and dialect Arabic. Each example shows the desired output format and medical reasoning level.

Challenges and solutions.

- Different formats: One prompt with high-quality examples and clear output rules ensures consistency across types without fine-tuning.
- Arabic language complexity: Carefully chosen examples and consistent decoding reduce errors and inconsistencies.
- Safety/clinical tone: The prompt guides toward brief, careful advice and marks cases needing doctor follow-up.

Reproducibility notes. Use the exact prompt templates provided in Appendix B; keep the examples unchanged; do minimal, consistent output cleaning as specified above. All runs use Gemini 2.5 Flash with the decoding settings specified in Table A.4.

4 Experimental Setup

4.1 Data and Splits

We follow the official AraHealthQA-2025 Track 2 (MedArabiQ) setup and evaluated directly via the organizers' API on the official test sets (ST1: 100 items, ST2: 100 items). The provided development sets (ST1: 300 items; ST2: 400 items) were used only to guide prompt design, select few-shot examples, and perform sanity checks. No fine-tuning or external training data was used.

4.2 Preprocessing

We applied only input-side, minimal steps to ensure consistent prompts and data cleanliness:

- Standardize Arabic punctuation and whitespace in the input text while preserving medical terminology and numbers.
- Normalize option labels and bullet symbols in MCQ questions to a consistent form before prompting.

4.3 Post-processing

We applied lightweight output-side normalization for evaluation stability:

- MCQ: map any predicted symbol to the canonical set {أ، ب، ج، د، هـ} and strip extra tokens.

- Generation: remove markdown (bold/italic/bullets), standardize commas and spaces, and keep a concise clinical tone.

Removing markdown formatting from generated text is essential, as structured formatting can introduce noise that affects evaluation metrics like BERTScore by altering token representations (Tang et al., 2024).

4.4 Prompting Configurations

For Sub-Task 1, we use three complementary prompts: Arabic Few-Shot (AFS), English Translation + Answer (ETA), and Refinement + Answer (RFA). Predictions are combined via simple majority vote with a fixed tie-breaker (RFA > AFS > ETA). For Sub-Task 2, a single unified Arabic instruction with few-shot examples handles fill-in-the-blank, patient–doctor Q&A, GEC, and paraphrased inputs.

4.5 Evaluation Metrics

- Sub-Task 1 (MCQ): Accuracy
- Sub-Task 2 (Generation): BERTScore

5 Results

We present our official results from the AraHealthQA-2025 shared task evaluation , analyzing performance across both subtasks and examining the effectiveness of our ensemble approach.

5.1 Sub-Task 1: Classification Results

Our ensemble approach achieved 76% accuracy on the official test set, securing 2nd place in the classification task. Table 1 presents detailed performance breakdown for each individual approach and the final ensemble.

Individual system performance. The Refinement + Answer (RFA) approach performed best among individual systems at 74% accuracy, demonstrating the effectiveness of question clarification and option explanation in Arabic medical contexts. The Arabic Few-Shot (AFS) approach achieved 71% accuracy, showing strong baseline performance with domain-specific examples. The English Translation + Answer (ETA) approach scored 69% accuracy, indicating some information loss during translation despite maintaining medical terminology.

Ensemble effectiveness. The 3-system ensemble (RFA + AFS + ETA) improved performance by 2 percentage points over the best individual system, reaching 76% accuracy. This demonstrates successful bias reduction through diverse prompt strategies, with the RFA approach providing clarity, AFS maintaining Arabic medical context, and ETA offering cross-lingual reasoning perspectives.

5.2 Sub-Task 2: Generation Results

Our unified prompting approach achieved 86.953% BERTScore on the official test set, securing 2nd place in the generation task. The approach used a single Arabic instruction prompt with few-shot examples, casting the model as an Arabic medical expert to handle diverse question formats including fill-in-the-blank, patient-doctor consultations, grammatical error correction, and paraphrased questions. This unified strategy proved effective across all question types without requiring task-specific fine-tuning, demonstrating the power of well-designed prompting for Arabic medical contexts. Table 2 summarizes the performance.

5.3 Ablation Studies

Ensemble composition. Removing individual systems from the 3-way ensemble showed: RFA removal (-3% accuracy), AFS removal (-2% accuracy), ETA removal (-1% accuracy), confirming the value hierarchy and ensemble complementarity.

Post-processing impact. Arabic text normalization and markdown removal improved Sub-Task 2 BERTScore by approximately 2-3%, demonstrating the importance of output standardization for evaluation metrics.

6 Conclusion

We presented a compact, prompt-engineering-based pipeline for Arabic clinical QA that performs robustly across diverse formats without fine-tuning. A small ensemble improves Sub-Task 1 classification, while a unified instruction guides Sub-Task 2 generation. Future extensions include retrieval augmentation with vetted Arabic medical sources, broader model diversity, and human-in-the-loop validation to mitigate ambiguity and domain gaps.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi.

Table 1: Sub-Task 1 (Classification) official results on test set. All experiments used Gemini 2.5 Flash.

Approach	**Description**	**Accuracy (%)**	**Ranking**
English Translation (ETA)	Translate to English then answer	69.0	–
Arabic Few-Shot (AFS)	Arabic instruction + 6 medical examples	71.0	–
Refinement + Answer (RFA)	Question clarification + option explanation	74.0	–
Ensemble (Final)	**3-way majority vote (RFA + AFS + ETA)**	**76.0**	**2nd**

Note: Individual systems not submitted separately; ensemble represents official submission.

Table 2: Sub-Task 2 (Generation) official results on test set using unified prompting approach.

Approach	**Description**	**BERTScore (%)**
Unified Arabic Prompting	Single prompt with Arabic medical expert role-playing, few-shot examples, handles all question formats	**86.953**
Final Ranking	**Official AraHealthQA-2025 shared task**	**2nd place**

BERTScore combines BLEU, ROUGE, and semantic similarity metrics.

2021. Arbert & marbert: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4510–4521.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E Shamout. 2025. Medarabiq: Benchmarking large language models on Arabic medical tasks. *arXiv e-prints*, pages arXiv–2505.

Youssef Al Hariri and Ibrahim Abu Farha. 2024. SMASH at StanceEval 2024: Prompt engineering LLMs for Arabic stance detection. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 800–806, Bangkok, Thailand. Association for Computational Linguistics.

Ashwag Alasmari, Sarah Alhumoud, and Waad Alshammari. 2024. Aramed: Arabic medical question answering using pretrained transformer language models. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, pages 50–56.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, and 1 others. 2025. Arahealthqa 2025 shared task description paper. *arXiv preprint arXiv:2508.20047*.

Jennifer Chu-Carroll, Krzysztof Czuba, John Prager, and Abraham Ittycheriah. 2003. In question answering, two heads are better than one. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 24–31.

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)*, pages 4171–4186.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9459–9474.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-Bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 12–34, Mexico City, Mexico. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V

Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

A Tables

This appendix contains tables referenced in the main text.

A.1 Classification Examples

Table 3: Examples on classification problem (Sub-Task 1).

Type	Inputs	Outputs
Multiple Choice Questions	كل ما يلي صحيح عن السفلس ماعدا: أ. يتميز الطور الأول بسلبية الاختبارات المصلية، ب. يتميز الطور الأول بقرحة صلبة مؤلمة على الأعضاء التناسلية، ج. يتميز الطور الثاني باندفاعات على الجلد والأغشية المخاطية، د. 25% من الأجنة تموت بعد الولادة من أم مصابة، هـ. يعاني الطفل المصاب بالزهري الخلقي من أسنان هوتشنسن.	ب
Fill-in-the-blank with choices	الحمرة هي عدوى جلدية تسببها ______، وهي تصيب عادةً ______ الوجه. أ. المكورات العنقودية الذهبية، البشرة، ب. العقديات B و C، الأدمة الشبكية، ج. العقديات A و G، الأدمة الحليمية، د. العصيات سلبية الجرام، الأنسجة تحت الجلد.	ج

A.2 Generation Examples

Table 4: Examples on generation problem (Sub-Task 2).

Type	Input	Output
Fill-in-the-blank	الحمرة هي عدوى جلدية تسببها ______، وهي تصيب عادةً ______ الوجه.	العقديات A و G، الأدمة الحليمية
Patient-Doctor Q&A	انا امرأة عمري 24 سنة، اشعر بألم ف بطني شديد الألم اشعر بعصر ف بطن و مغص و غثيان و فقدان شهيه...	يرجى عمل تحليل البراز وموافتنا بالنتيجة لتحديد العلاج...
Grammatical Error Correction (GEC)	انا امرأة عمري 24 سنة، اشعر بألم ف بطني شديد الألم اشعر بعصر ف بطن و نعرات ف البطن...	يرجى عمل تحليل البراز وموافتنا بالنتيجة لتحديد العلاج...
LLM Paraphrasing	انا امرأة عمري 24 سنة، لدي ألم في بطني مصحوب بمغص وغثيان وفقدان للشهية...	يرجى عمل تحليل البراز وموافتنا بالنتيجة لتحديد العلاج...

A.3 Data Refinement Examples

Table 5: Data refinement examples showing improvements in question clarity and formatting.

Version	Issue	Question Text
Original	Unclear formatting	في التهاب المشيمية نصادف الأشكال الالتهابية التالية: (الخاطئة) أ. التهاب مشيمية نتحي ب. التهاب مشيمية منتشر ج. التهاب مشيمية أمامية د. التهاب مشيمية مركزي ه. التهاب مشيمية زاوي
	Ambiguous phrasing	كل ما يخص النيجيرية الدجاجية صحيح ما عدا: أ. تعيش هذه المتحولة بشكل حر في الماء والتربة ب. تسبب حالات التهاب سحايا ودماغ بدئي العائل الناقل الذباب ج. يعيش هذا الطفيلي في المياه المعدنية الساخنة ناقصة الأكسجة د. تعتبر النيجلرية التجاجية أسرع وأشد إمراضية من الشوكيية
Refined	Clear formatting	في التهاب المشيمية نصادف الأشكال الالتهابية التالية، ما عدا: أ. التهاب مشيمية نتحي (التهاب موضعي محدود في منطقة معينة) ب. التهاب مشيمية منتشر (التهاب يشمل مناطق واسعة) ج. التهاب مشيمية أمامي (التهاب في الجزء الأمامي من المشيمية) د. التهاب مشيمية مركزي (التهاب في المنطقة المركزية) ه. التهاب مشيمية زاوي (مصطلح غير دقيق طبياً)
	Enhanced clarity	كل ما يلي صحيح عن النيجلرية الدجاجية ما عدا: أ. تعيش بشكل حر في الماء والتربة (كائن حي مجهري حر المعيشة) ب. تسبب التهاب السحايا والدماغ الأولي (عدوى خطيرة في الجهاز العصبي) ج. العائل الناقل هو الذباب (معلومة خاطئة - لا ينتقل عبر الذباب) د. تعيش في المياه المعدنية الساخنة قليلة الأكسجة (بيئة خاصة للنمو) ه. أسرع وأشد إمراضية من الشوكيية (خصائص مرضية مميزة)
Fill-in-blank	Missing context	نستخدم أغشية مصنوعة من ______ في تقنيات التبقيع. أ. النايلون أو السيللوز ب. site acyl Amino ج. site Peptide د. الاسيتونتريل مع مادة TEAA
	Clear context	في تقنيات التبقيع المخبرية، نستخدم أغشية مصنوعة من ______: أ. النايلون أو السيللوز (مواد ماصة للبروتينات) ب. acyl Amino site (موقع ربط الأحماض الأمينية) ج. site Peptide (موقع تكوين الببتيدات) د. الأسيتونيتريل مع TEAA (مذيبات كروماتوغرافية)

A.4 Hyperparameters

Table 6: Decoding hyperparameters used for all experiments with Gemini 2.5 Flash.

Parameter	Value
Temperature (τ)	0.1
Top-p	0.8
Top-k	40

B Prompt Templates

This appendix contains the complete prompt templates used in our experiments for reproducibility.

Table 7: Complete prompt templates used in Sub-Task 1 and Sub-Task 2.

Prompt Type	Template
Arabic Few-Shot (AFS)	أنت مساعد طبي خبير وموثوق للغاية. مهمتك هي الإجابة بدقة لا متناهية على الأسئلة الطبية المقدمة باللغة العربية، مع الالتزام التام بتنسيق الإجابة المطلوب. نوع الأسئلة التي ستتلقاها: أسئلة الاختيار من متعدد: تتضمن سؤالاً وخيارات إجابة مرقمة بأحرف عربية (أ، ب، ج، د،). أسئلة إكمال الفراغ: تتضمن جملة أو فقرة بها فراغ واحد أو أكثر، وتُتبع بخيارات إجابة مرقمة. Few-shot examples: المثال 1 (علم الأدوية): السؤال: هـ. لا يجوز مشاركة الكازولين مع البكتين. الإجابة الصحيحة: هـ [... 5 more examples] التعليمات الأساسية للإجابة: 1. الفهم الشامل: اقرأ السؤال وجميع الخيارات المتاحة بعناية فائقة. 2. استخدام المعرفة: استعن بمعرفتك العميقة والموثوقة في المجالات الطبية. 3. تحديد الإجابة الصحيحة: اختر الخيار الأنسب. 4. صيغة الإجابة المطلوبة (صارمة): يجب أن تكون إجابتك حرفاً عربياً واحداً فقط.
Translation (ETA)	You are a medical translation expert. Translate the following Arabic medical question into English following these exact requirements: 1. Maintain the medical accuracy and terminology 2. Format the question properly with options A, B, C, D, E 3. Use "**except**" formatting when the question asks for the wrong/false option 4. Keep the medical context and meaning intact 5. Use proper English medical terminology
Refinement (RFA)	أنت خبير في الطب وتحرير النصوص الطبية. مهمتك هي تحسين وضوح وسلاسة الأسئلة الطبية التالية باللغة العربية مع الحفاظ على: 1. المعنى الطبي الدقيق 2. تنسيق الخيارات (أ، ب، ج، د، ه) 3. الفراغات للأسئلة من نوع "املأ الفراغ" 4. الأرقام والرموز العلمية 5. المصطلحات الطبية باللغة الإنجليزية كما هي. مطلوب إضافي: أضف شرحاً مختصراً (15-25 كلمة) لكل خيار من الخيارات لتوضيح المعنى الطبي.
Generation (Sub-Task 2)	أنت طبيب خبير ومستشار صحي موثوق، ومتخصص في تقديم إجابات طبية دقيقة ومحترفة باللغة العربية. مهمتك هي الإجابة على استفسارات طبية متنوعة، تتراوح بين إكمال الفراغات والرد على استشارات المرضى. التعليمات الأساسية: 1. التحليل الدقيق: اقرأ السؤال أو الاستشارة بعناية فائقة لفهم السياق الطبي المطلوب. 2. استحضار المعرفة: استخدم معرفتك المتعمقة في الطب والعلوم السريرية. 3. صيغة الإجابة المطلوبة: لأسئلة إكمال الفراغ: أجب فقط بالكلمة أو الكلمات المطلوبة. للاستشارات الطبية المفتوحة: قدم إجابة مباشرة ومفيدة. 4. الالتزام بالمصطلحات: استخدم المصطلحات الطبية الصحيحة باللغة العربية. 5. التجنب: لا تكتب أي تفسير أو شرح إضافي. 6. التخصيص: انتبه للمعلومات التي تخص المريض.

Sindbad at AraHealthQA Track 1: Leveraging Large Language Models for Mental Health QA

AbdulRahman A. Morsy[*1], **Saad Mankarious**[*1], **Aya Zirikly**[1,2]
[1]Department of Computer Science, The George Washington University (USA)
[2]Center for Language and Speech Processing (CLSP), Johns Hopkins University (USA)
{abdulrahman.morsy, saadm, ayah.zirikly}@gwu.edu

Abstract

Mental health detection in online discourse is a growing area of NLP research, particularly for low-resource languages such as Arabic, where stigma and limited access to professional care make anonymous, technology-driven solutions valuable. In the context of the AraHealth shared task, we were provided with three subtasks: multi-label classification for questions, multi-label classification for answers, and a QA system leveraging models developed in the previous two tasks. Our approach employed data augmentation to address class imbalance, as certain categories in the dataset were significantly underrepresented. Since our method relied on prompting models to classify questions and answers as well as to generate answers for the QA system, we utilized Gradient-free Edit-Based Instruction Search (GrIPS) to optimize prompt selection. Our system achieved strong results across all three subtasks, ranking 1st in answer classification and 3rd in both question classification and QA system answer generation.

1 Introduction

Mental health issues are a global concern with substantial economic and social impact (Santomauro et al., 2021). This challenge is particularly pronounced in Arabic-speaking communities, where discourse around mental health remains stigmatized, and access to professional resources is often limited or treated as a luxury (Khatib et al., 2023). Such constraints motivate NLP research that can detect and address mental health concerns using online data (Zirikly et al., 2019; Shing et al., 2018), enabling more anonymous, unrestricted, and accessible support tools for individuals in the Arab world (Hassib et al., 2022).

In this shared task (Alhuzali et al., 2025), we were provided with a dataset curated from an Arabic-language online forum that follows a question–answer (QA) pattern between patients and mental health professionals (Alhuzali et al., 2024). The dataset comprises 350 annotated instances, each containing a question, its corresponding answer, and categorical labels. Specifically, every sample is assigned one or more labels from seven possible question categories, as well as one or more labels from three possible answer categories, thereby constituting a multi-label classification setting. The distribution of these categories is highly imbalanced, especially for questions, which motivated our data augmentation approach. We employed GPT to generate additional samples for minority classes, resulting in a more balanced dataset. Using the augmented data, we performed instruction fine-tuning with a range of pre-trained models. In parallel, we explored few-shot prompting for multi-label classification in Task 1 (question classification) and Task 2 (answer classification). For Task 3 (QA system), we leveraged the fine-tuned models from the first two tasks and applied Gradient-free Edit-Based Instruction Search (GrIPS) to optimize the prompts used for answer generation.

Our system achieved strong results across all tasks: we ranked 1st in answer classification (Sub-Task 2) and 3rd in both question classification (Sub-Task 1) and QA answer generation (Sub-Task 3).

2 Background

The shared task (Alhuzali et al., 2025) focused on three subtasks in the Mental Health track: (1) multi-label question classification, (2) multi-label answer classification, and (3) a QA system for generating appropriate answers using models from the first two tasks. Each instance in the dataset consists of a question, its corresponding answer, a question category (one of seven possible labels), and an answer category (one of three possible labels). For

[*]Equal contribution

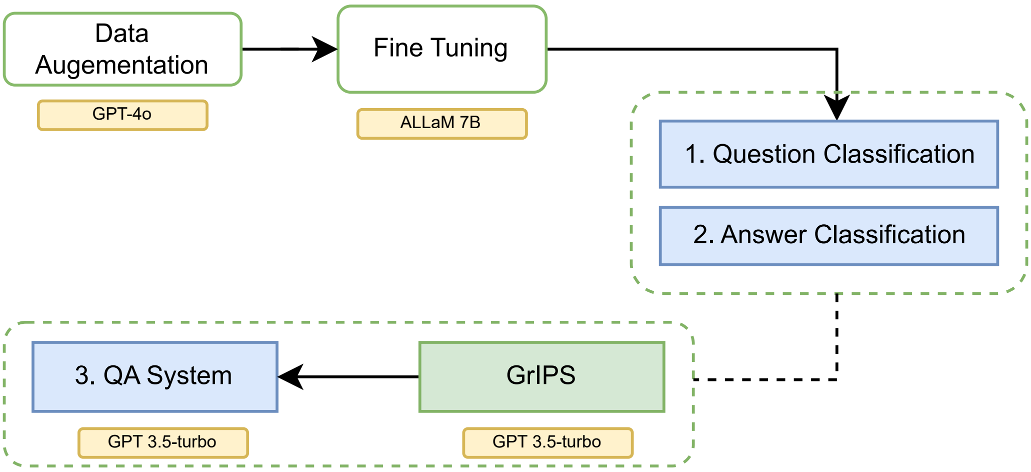


Figure 1: Overall pipeline for our approach. Beginning with the raw data, we generate synthetic samples and leverage them to perform classification and then answer generation.

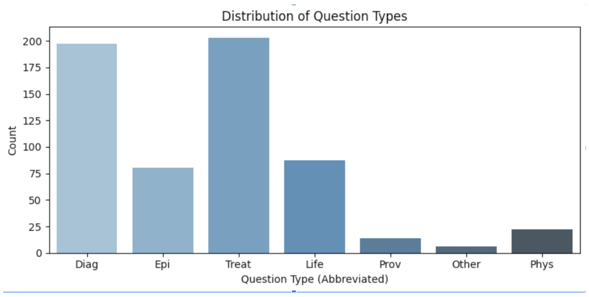


Figure 2: Question class distribution, showing significant imbalance between categories.

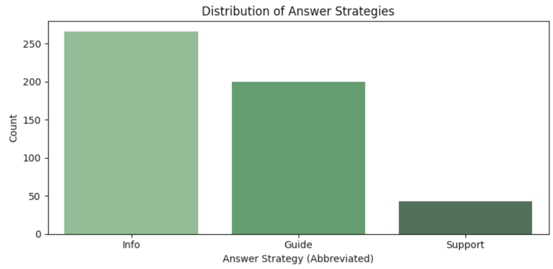


Figure 3: Answer class distribution.

example, a question about treatment options for depression could be labeled under *Treatment* for questions and *Supportive Advice* for answers.

The question taxonomy spans seven categories, covering clinical reasoning and practical guidance: Diagnosis (A) for interpreting findings, Treatment (B) for therapeutic options, Anatomy and Physiology (C) for biomedical knowledge, Epidemiology (D) for disease progression and causes, Healthy Lifestyle (E) for wellness habits, Provider Choices (F) for healthcare navigation, and Other (Z) for miscellaneous queries. Answer strategies fall into three broad types: Information (1) delivering factual content and resources, Direct Guidance (2) offering actionable recommendations, and Emotional Support (3) providing reassurance or encouragement (Alhuzali et al., 2024).

The dataset contains 350 labeled QA pairs from an Arabic-language mental health forum. Figures 2 and 3 show the category distributions. The question categories are heavily imbalanced, with certain categories having fewer than 10 samples, while the largest category has over 175 samples. The answer categories are also imbalanced, though less severely. This imbalance strongly motivated our data augmentation approach to generate synthetic samples for minority classes.

Our participation covered all three tracks, and our contribution is novel in its integration of prompt optimization (via GrIPS) with both few-shot prompting and instruction fine-tuning for imbalanced and low-resource Arabic mental health classification tasks. Related work in Arabic NLP has explored mental health (Alhuzali and Alasmari, 2025, 2024), but to our knowledge, no prior shared task system has combined prompt optimization with synthetic minority oversampling for both classification and answer generation in this domain. For instance, MedArabiQ (Abu Daoud et al., 2025) introduced a benchmark for evaluating large language models on Arabic medical tasks, covering a wide range of QA problems.

3 System Overview

We built the QA system by leveraging our models from Subtasks 1 and 2, which classify questions and answers. As shown in Figure 4, we appended the predicted question category to the QA prompt. We used the question category predicted labels (obtained from fine-tuning) as guidelines for the QA model. Furthermore, the system prompt is optimized using GrIPS as Section 3.3 explains.

3.1 Data Augmentation

To address dataset size constraints and class imbalance, we employed GPT-4o (temperature = 0.7) to synthesize additional training instances. Augmentation targeted the least frequent labels: (3) and its multi-label variants (1,3), (2,3), and (1,2,3) for Subtask 2; and labels C, D, E, and F for Subtask 1.

Prompt construction incorporated: (i) role specification to enforce domain-appropriate tone; (ii) category definitions from (Alhuzali et al., 2024); (iii) in-context exemplars; (iv) explicit formatting constraints (e.g., fixed sample counts, variable lengths); and (v) lexical variation controls to minimize redundancy.

For Subtask 1, we generated 300 synthetic samples (50 each for D and E; 100 each for C and F), expanding the dataset from 350 to 650 samples. For Subtask 2, we generated 160 samples (40 per target configuration), expanding the dataset from 350 to 510 instances.

Following the generation of each set of samples, we performed a human evaluation to assess the quality and relevance of the generated samples. For this purpose, we randomly selected approximately one-third of each set for detailed inspection with respect to fluency, label relevance, and adherence to the specified constraints.

Appendix A.1 includes examples of both the prompts and the generated data. Additionally, the full datasets, including all generated samples, are available on our GitHub repository.[1]

3.2 Model Fine-Tuning

We used fine-tuned models from Subtask 1 (Question Classification) to explicitly solve Task 1. Furthermore, we leveraged the best performing model (ALLaM 7B) to develop the QA system as shown in Figure 1. Given a question that the system needs to respond to, we obtain a predicted category label from the fine-tuned models that we then provide in the prompt. This is the second step in our prompt building process in Figure 4.

3.3 Prompt Optimization with GrIPS

Gradient-free Instructional Prompt Search (GrIPS) is a technique proposed by (Prasad et al., 2022) to efficiently optimize the prompts used for our QA system. We used GrIPS to optimize the system prompt of our QA System as Figure 4 shows. Our implementation of GrIPS follows an iterative prompt optimization process. Starting from an initial prompt, we generate candidate variations through targeted mutations, such as structural adjustments, content refinements, and cultural adaptations for six iterations. Each candidate prompt is evaluated on a subset of the training data using BERTScore F1 to measure the alignment between the model-generated and reference answers. See Figure 5 for BERTScore performance for each iteration. The highest-scoring prompt is retained for the next iteration, and the process is repeated for a fixed number of optimization rounds. This approach enables systematic improvement of prompt effectiveness without gradient-based updates, ultimately yielding an optimized instruction that enhances model performance on the QA system.

4 Experimental Setup

4.1 Sub-Task 1: Question Classification

Prior to building the QA system, we employed fine-tuning to classify questions. We experimented with an array of PLMs and used the best performing model (ALLaM 7B) in the QA System. The hyperparameters used are listed in Table 1. Table **??** in the appendix also shows the performance on the evaluation portion (20 percent) of our augmented dataset.

4.2 Sub-Task 2: Answer Classification

For Subtask 2, we fine-tuned **MARBERT** (Abdul-Mageed et al., 2021) on the augmented dataset described in Section 3.1 for multi-label classification, using the hyperparameters in Table 2, consistent with (Alhuzali and Alasmari, 2025) for the same task. Model training employed a maximum sequence length of 256, the *Binary Cross-Entropy* loss function, and an *AdamW* optimizer.

An 80/20 train-validation split was used for cross-validation to ensure that model performance was stable and not driven by outliers. Then, we

[1] https://github.com/AbdulRahmanBenatia/Sindbad-AraMentalQA-SharedTask

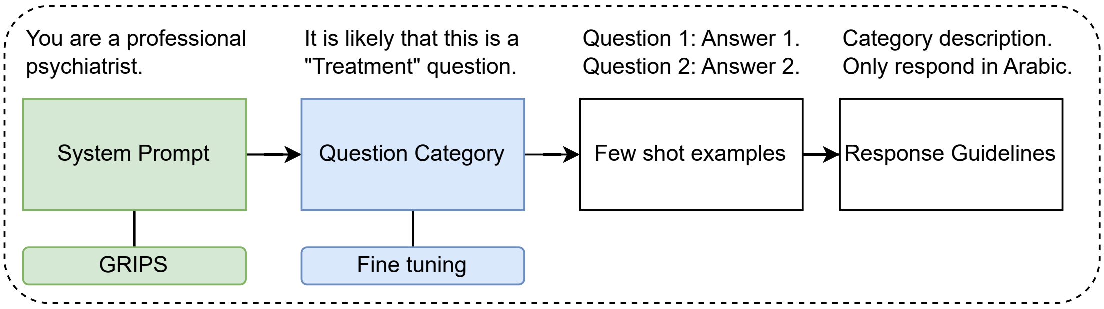


Figure 4: Overall prompt structure for the QA System.

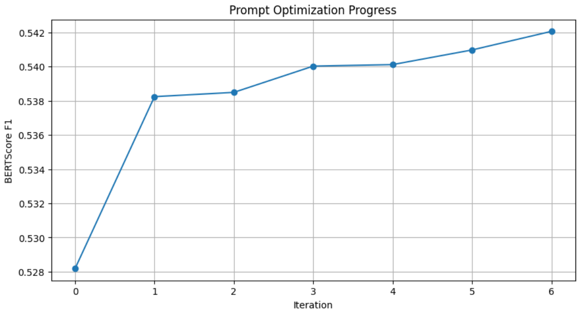


Figure 5: GPT-3.5-turbo Performance during GrIPS prompt optimization.

Parameter	**Value**
Learning rate	1×10^{-4}
Batch size	2
Gradient accumulation steps	4
Epochs	3
Weight decay	0.01
Early-stop patience	2
LORA rank (r)	8
LORA alpha	16
LORA dropout	0.05
Quantization	None
Optimizer	adamw_torch

Table 1: Default hyperparameters for fine-tuning on Subtask 1, question classification.

re-trained the model on the full augmented set and used it to generate predictions for the official test set. We report our results using the following evaluation metrics: *Weighted-F1* and *Jaccard* score, as recommended by the shared task organizers.

4.3 Sub-Task 3: Question Answering System

Parameter	**Value**
Hidden size	768
Batch size	8
Dropout	0.1
Early-stop patience	10
Epochs	15
Learning rate	2×10^{-5}
Optimizer	AdamW

Table 2: Hyperparameters for MARBERT fine-tuning.

We employed `gpt-3.5-turbo` via the OpenAI API for question–answer (QA) generation. The system adopted a few-shot prompting approach with three manually carefully selected examples that represent diversity in target labels. Categories were drawn from a predefined taxonomy (A–F, Z) covering diagnosis, treatment, anatomy/physiology, epidemiology, healthy lifestyle, provider choice, and miscellaneous queries.

Prompts were structured into: (i) a system role enforcing professional and empathetic Arabic medical responses; (ii) category-specific contextual descriptions; (iii) explicit response guidelines; and (iv) optional in-context examples. The temperature parameter was fixed at 0.0 to ensure deterministic output and reproducibility.

5 Results

5.1 Sub-Task 1

Our ALLaM-based system achieved third place in Subtask 1 with *Weighted-F1* $= 0.53$ and *Jaccard* $= 0.49$ as shown in Table 3.

Weighted-F1	**Jaccard**	**Ranking**
0.53	0.49	3rd

Table 3: Results on the official test set for Subtask 1.

5.2 Sub-Task 2

As shown in Table 4, our system achieved *Weighted-F1* $= 0.79$ and *Jaccard* $= 0.71$, ranking first in Subtask 2. The observed performance gains were primarily attributed to the targeted augmentation of under-represented label combinations.

Weighted-F1	Jaccard	Ranking
0.79	0.71	1st

Table 4: Results on the official test set for Subtask 2.

5.3 Sub-Task 3

On the official test set, our `gpt-3.5-turbo` system achieved a BERTScore of 0.668, ranking 3rd in Subtask 3. This performance reflects the benefit of structured, category-aware prompting and few-shot exemplars, though the gap to the top systems suggests potential for further domain adaptation.

BERTScore	Ranking
0.668	3rd

Table 5: Results on the official test set for Subtask 3.

6 Conclusion

In this work, we developed a system for Arabic mental health QA tasks that integrates instruction fine-tuning on augmented data, few-shot prompting, and gradient-free prompt optimization via GrIPS. Our approach effectively addresses class imbalance and low-resource challenges, achieving first place in answer classification (Sub-Task 2) and third place in both question classification (Sub-Task 1) and QA answer generation (Sub-Task 3). These results demonstrate the effectiveness of combining synthetic data generation, fine-tuning, and prompt engineering to enhance large language model performance in specialized, low-resource domains and open the way for further research addressing Arabic mental health NLP.

References

M. Abdul-Mageed, A. Elmadany, and E. M. B. 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, Virtual.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E Shamout. 2025. Medarabiq: Benchmarking large language models on arabic medical tasks. *arXiv e-prints*, pages arXiv–2505.

H. Alhuzali and A. Alasmari. 2025. Pre-trained language models for mental health: An empirical study on arabic q&a classification. *Healthcare*, 13(9):985.

H. Alhuzali, A. Alasmari, and H. Alsaleh. 2024. MentalQA: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*.

Hassan Alhuzali and Ashwag Alasmari. 2024. Evaluating the effectiveness of the foundational models for q&a classification in mental health care. *arXiv preprint arXiv:2406.15966*.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash, and Leen Kharouf. 2025. Arahealthqa 2025 shared task description paper. In *Proceedings of ArabicNLP 2025*.

Mariam Hassib, Nancy Hossam, Jolie Sameh, and Marwan Torki. 2022. AraDepSu: Detecting depression and suicidal ideation in Arabic tweets using transformers. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 302–311, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Hania El Khatib, Aisha Alyafei, and Madiha Shaikh. 2023. Understanding experiences of mental health help-seeking in arab populations around the world: A systematic review and narrative synthesis. *BMC Psychiatry*, 23(1):324.

Archiki Prasad, Pratyush Venkatesh, Yixin Xu, Rui Zhang, Junyi Jessy Li, Surya Kallumadi, and Bill Yuchen Lin. 2022. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*.

Damian F. Santomauro, Ana M. Mantilla Herrera, Junfang Shadid, Peng Zheng, Charlie Ashbaugh, Giorgia Pigott, Harish G. Sheena, and et al. 2021. Global prevalence and burden of depressive and anxiety disorders in 204 countries and territories in 2020 due to the covid-19 pandemic. *The Lancet*, 398(10312):1700–1712.

Han-Chin Shing, Suraj Nair, Ayah Zirikly, Meir Friedenberg, Hal Daumé III, and Philip Resnik. 2018. Expert, crowdsourced, and machine assessment of suicide risk via online postings. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pages 25–36, New Orleans, LA. Association for Computational Linguistics.

Ayah Zirikly, Philip Resnik, Özlem Uzuner, and Kristy Hollingshead. 2019. CLPsych 2019 shared task: Predicting the degree of suicide risk in Reddit posts. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 24–33, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix

A.1 Data Augmentation

Figure 6 presents an example prompt along with three randomly selected generated responses, illustrating Answer augmentations associated with labels (1, 3).

Similarly, Figure 7 presents an example prompt together with five randomly selected generated responses, illustrating Question augmentations associated with label E.

A.2 Fine-tuning

We experimented with the following arrays of models during fine tuning:

```
"meta-llama/Llama-2-13b-chat-hf",
"ALLaM-AI/ALLaM-7B-Instruct-preview",
"silma-ai/SILMA-9B-Instruct-v1.0",
"aubmindlab/bert-base-arabertv2",
"UBC-NLP/MARBERT",
"CAMeL-Lab/bert-base-arabic-camelbert-
    mix"
```

Model	F1 Micro	F1 Wtd.	Jaccard
meta-llama	0.62	0.57	0.47
ALLaM-7B	**0.68**	**0.59**	**0.54**
SILMA-9B	0.62	0.57	0.48
arabertv2	0.68	0.59	0.53
MARBERT	0.68	0.59	0.53
camelbert-mix	0.68	0.59	0.53

Table 6: Performance of different models on the evaluation set for Subtask 1: Question Classification.

A.3 GrIPS

To optimize the prompt used in the QA system, we ran the optimization script against the initial template, which resulted in the final prompt header

```
#Initial Prompt Header
"You are an expert Arabic mental health
    assistant.
    Provide accurate, helpful responses
        to mental health questions in
        Arabic.
    Be professional yet empathetic in
        your answers."

#Final Prompt Header
"You are an expert Arabic psychiatric
    conditions assistant.
    Provide accurate, evidence-based
        responses to psychiatric
        conditions questions in Arabic.
    Be professional yet empathetic in
        your answers. Answer concisely
        in Arabic medical terminology.
        Answer concisely in Arabic
        medical terminology."
```

Note: this represents the prompt header, often supplied as a system propmpt to applicable models. In the rest of the prompt, we append the categories description as given the description of the task, along with other information to control the model response.

أنت طبيب نفسي بارع ذو شهرة واسعة وتحب إجابة أسئلة المرضى بدقة وخبرة.

لديك ثلاثة أنواع من الإجابات على أسئلة المرضى، منها ما يلي:

Information. This category includes answers that provide information, resources, etc. It also includes requests for information.

Emotional Support. This category includes answers that provide approval, reassurance, or other forms of emotional support.

أمثلة: واضح انك توتري قوي حاولى انك وأنت بتتكلمى مع الآخرين انك لا تأخذي الموضوع على انك في وضع تقييم ثقي في نفسك و ركزي عليها اكتر من رأي الناس فيك

سلامتك الوسواس مرض نفسي قابل للعلاج وكلما كان مبكرا اصبح ايسر في العلاج

انت حاسس ان ده افضل حل حاليا لان مفيش حاجة حلوة في حياتك وكل يوم بتعيشه مؤلم بالنسبة لك، طيب لو في فرصة تطلب المساعدة وتلاقي حاجات حلوة واهداف مهمة في حياتك، مش يمكن ده يخليك تعيد النظر في اختيارك، ارجو التواصل معايا عن طريق رقم التلفون يمكن اقدر اساعدك

في حالات اضطراب ثنائي القطب لا ينصح باستخدام مضادات الاكتئاب بشكل عام (مودابكس،سيربراس)..الافضل الالتزام بخطتك العلاجية من قبل طبيبك المعالج.و في حال ظهور اي اعراض للانتكاسة مراجعة الطبيب للتقييم و اجراء اللازم…الف سلامة عليك.

هل يمكنك إعطاء الكثير من النماذج لهذا تقتصر على الدعم والمعلومات فقط دون إرشاد علاجي. فقط دعم ومعلومات بخبرة الطبيب النفسي البارع صاحب اللغة الواضحة.

رجاء أعطني على الأقل 40 إجابة من هذا النوع لأسئلة مختلفة، وليقتصر ردك على الإجابات مرقمة من 1 إلى 40 وبينها سطر فارغ.

اجعل طول الإجابات متنوعًا أي لا تقتصر على الجمل القصيرة أو الطويلة بل أريد تنوعًا واضحا.

(a) Example prompt.

- واضح من كلامك إنك تعاني من توتر مزمن. التوتر له تأثير كبير على الجسم والنفس، ويمكن أن يظهر في شكل صداع، أرق، أو حتى آلام عضلية. مجرد إدراكك للمشكلة هو خطوة كبيرة نحو التحسن. لا تحمل نفسك فوق طاقتها، وأنت تستحق أن ترتاح.
- الوسواس القهري مرض نفسي شائع، وله أسباب بيولوجية ونفسية. يتميز بأفكار متكررة وسلوكيات قهرية يصعب التحكم بها. وجودك هنا وسؤالك يدل على وعي ورغبة في التغيير، وهذا شيء مشجع جدًا.
- كثير من الناس يمرون بما تمر به، والشعور بالحزن لفترات طويلة قد يكون مرتبطًا بأعراض اكتئابية. ما تحسه حقيقي، وليس مبالغة، وأنت تستحق أن يُفهم ويُحتوى.

(b) Sample responses.

Figure 6: Augmentation prompt and sample generated responses for Answers (label 1, 3).

أنت مولع بعلم النفس والطب النفسي، تقرأ كثيرًا أسئلة المرضى وإجابات الأطباء عليها. أنواع الأسئلة متعددة، منها النوع الآتي:

Healthy lifestyle: Questions are specified to diet, exercise, mood control and other lifestyle factors that can affect health.

أمثلة:

انا غير راضية عن نفسي من نايحة المظهر اصبحت لدي عقدة اشعر اني بشعة والله انا اخجل من نفسي اصبحت دائما اقارن نفسي بالبنات الاخريات واحسدهن على جمالهن كثيرا ما تأحسر على مضهري انعدمت ثقتي بنفسي ودائما انضر الى المراة مع العلم اني لست بدالك السوء انا عادية اغلب الناس من يقول اني جميلة لكن لا اصدقهم مع العلم انه ليس لدي اي تشوهات معدا مستوى الاسنان احينا ما اتعرض لسخرية من اخوتي والله اشعر بالاحباط شديد

انا رجل ابلغ من العمر اربعون عام ولم اتزوج من قبل ولكن الان افكر في الزواج والبعض ينصحني بعدم الزواج لتقدم سني واني لا استطيع ان اعيش حياة سعيده بعد هذا العمر

حزن الرغبه فنوم لفتره طويله لا اريد التكلم والحديث

هل يمكنك إعطائي 50 سؤالًا آخر من هذا النوع. رجاء التزم هذا النوع، مع التنوع في الموضوعات وطول السؤال فبعض الأسئلة قصير وبعضها طويل وهكذا.

ليس ضروريًا أن تكون جميعها أسئلة، فبعضها أسئلة، وبعضها شكوى يشتكيها المريض (مع التزام النوع).

(a) Example prompt.

- فقدت رغبتي في الخروج من المنزل وأصبحت أفضل العزلة طوال الوقت.
- أشعر أنني أتناول الطعام طوال الوقت دون سبب حقيقي، فقط للهروب من التوتر.
- هل المشي اليومي كافٍ لتحسين المزاج والنوم؟
- لم أعد أستمتع بشيء، حتى الهوايات التي كنت أحبها أصبحت مملة.
- هل قلة شرب الماء تؤثر على التركيز والنشاط؟

(b) Sample responses.

Figure 7: Augmentation prompt and sample generated responses for Questions (label E).

Arabic Mental Health Question Answering: A Multi-Task Approach with Advanced Retrieval-Augmented Generation

AbdelAziz Amr **Mamdouh Koritam** **Mohamed Youssef**
Marwa Aldeeb Ensaf H. Mohamed
CIS, School of Information Technology and Computer Science, Nile University
Giza, Egypt
{a.amr2150, m.mohamed2158, m.ahmed2148, maldeeb, enmohamed}@nu.edu.eg

Abstract

Arabic-speaking communities face persistent challenges in mental health support due to linguistic complexity, cultural nuances, and limited specialized resources. This study introduces AraHealthQA 2025, a multi-task framework for Arabic mental health question answering, tackling three subtasks: (i) question classification, (ii) answer strategy classification, and (iii) generative question answering using a Retrieval-Augmented Generation (RAG) pipeline. For classification,fine-tuned AraBERTv2, MARBERTv2, and Arabic RoBERTa on multi-label mental health data. For generation, developing a culturally-aware RAG system that integrates semantic chunking, query enhancement, and hybrid retrieval. Dense retrieval via akhooli/Arabic-SBERT-100K, sparse retrieval via rank_bm25, and generation using Sakalti/Saka-14B fine-tuned with culturally aligned mental health terminology (e.g., respecting religious sensitivities in advice). The approach achieves weighted F1-scores of 0.742 (question classification) and 0.718 (answer classification), and a BERTScore F1 of 0.821 representing up to 15% improvement over retrieval-only baselines. These findings demonstrate the potential of culturally sensitive, Arabic-focused NLP systems to advance accessible mental health support.

1 Introduction

Imagine a young Arabic speaker in a rural Egyptian town seeking help online for anxiety. They describe their symptoms using local dialect and everyday expressions, but most automated systems either fail to understand the meaning or respond with advice that feels culturally inappropriate sometimes even contradicting religious or social norms. This reality reflects the urgent need for mental health question answering (QA) systems that understand both the Arabic language and the cultural context in which it is used.

Mental health support is a global challenge, yet it is particularly acute in Arabic speaking regions, where cultural stigma, linguistic diversity, and limited access to professional services create significant barriers to care. According to the World Health Organization, fewer than 30% of individuals in these countries receive adequate mental health support. Intelligent, culturally aware QA systems could help bridge this gap by making reliable, contextually appropriate information more accessible.

Arabic Natural Language Processing (NLP) in healthcare faces unique challenges: morphological complexity, dialectal variation, and scarcity of domain specific resources. Unlike English, where large scale datasets and specialized resources are abundant, Arabic mental health NLP suffers from a shortage of annotated datasets, sensitivity to cultural and religious norms, and the need for responses that reflect socially acceptable language and tone.

While transformer based models such as AraBERT and MARBERTv2 have demonstrated strong results in various Arabic NLP tasks, their application in mental health contexts particularly in multi-task frameworks remains largely unexplored. Moreover, existing Arabic QA systems rarely integrate mechanisms for cultural sensitivity, such as avoiding taboo topics, respecting religious guidelines in therapeutic advice, and translating formal medical terms into locally understood idioms. This work addresses the AraHealthQA 2025 workshop's Track 1: MentalQA challenge, which involves three interconnected tasks essential for a comprehensive mental health support system. Contributions are as follows:

(1)Multi-task Framework: A unified pipeline for question categorization, answer strategy classification, and generative QA, allowing better alignment between classification and generation.

(2)Advanced Arabic RAG System: A Retrieval Augmented Generation architecture optimized for

Proceedings of The Third Arabic Natural Language Processing Conference, pages 192–197
November 8-9, 2025 ©2025 Association for Computational Linguistics

Arabic mental health contexts, incorporating semantic chunking, query enhancement, and culturally aware reranking, which improves the relevance and appropriateness of generated responses.

(3)Comprehensive Evaluation: Extensive experimental analysis of transformer models in Arabic mental health classification tasks, supported by domain specific and semantic similarity metrics.

(4)Cultural Sensitivity Integration: Mechanisms to avoid inappropriate advice in sensitive contexts, for example, rephrasing lifestyle recommendations to respect religious fasting periods or reframing advice using culturally accepted idioms.

By addressing both the technical and cultural dimensions of the problem, this research provides a foundation for building Arabic mental health QA systems that are accurate, contextually aware, and socially responsible.

2 Related Work

2.1 LLMs in Healthcare

Large Language Models (LLMs) have been increasingly adopted in healthcare for tasks such as clinical decision support, diagnostics, and patient communication. Recent scopings highlight both their promise and the need for responsible integration, emphasizing ethical guidelines, transparency, and interdisciplinary collaboration (1). In mental health care specifically, research show applications in screening, symptom detection, conversational agents, and intervention support, while cautioning about hallucinations, bias, and reliability issues (2).

2.2 LLMs in Mental Health

Recent systematic studies report LLM applications in detecting depression, suicide risk, and delivering counseling or educational interventions (2). introducing *PsyLLM*, a specialized model integrating diagnostic and therapeutic reasoning aligned with DSM and ICD frameworks, which demonstrated improvements in realism, safety, and comprehensiveness compared to conventional LLMs (3).

2.3 RAG in Mental Health

Retrieval Augmented Generation (RAG) has been applied to enhance mental health recommendation systems. Evaluating baseline LLMs (GPT-3.5, GPT-4o, Gemma 2, Claude 4) for mental health app recommendations and found that while baseline models achieved 60–75% accuracy, RAG enhanced pipelines achieved 100% accuracy with improved diversity and quality (4).

2.4 Arabic Mental Health Applications

Arabic mental health NLP remains a developing field. Introducing the *MentalQA* dataset for Arabic mental health Q&A classification, showing that transformer based models like MARBERT outperform classical baselines, with GPT-3.5 few-shot prompting yielding notable accuracy improvements (5). Benchmarking multiple mono and multilingual LLMs for Arabic mental health support, finding that structured prompts improved performance by 14.5% on average, and few-shot learning boosted accuracy by 1.58× for certain models such as GPT-4o Mini (6).

Despite notable advances, several key gaps remain in Arabic mental health NLP. First, there is a lack of large scale, culturally aligned datasets for Arabic mental health QA. Second, few systems adopt multi-task approaches that integrate classification and generation for coherent end to end performance. Third, cultural integration is insufficient, with some systems producing outputs that conflict with social and religious practices (e.g., dietary advice during Ramadan). Finally, no fully developed, culturally aware Arabic RAG pipelines currently exist for this domain

3 Methodology

The proposed Advanced Retrieval Augmented Generation (RAG) System for Arabic Mental Health Q&A represented in figure 1. This pipeline integrates dense and sparse retrieval for Arabic mental health Q&A. Input Q&A data is chunked, embedded with Arabic-SBERT-100K, and indexed using both vector and BM25 methods. A user query is enhanced, retrieved, re-ranked, and passed along with the most relevant contexts to a finetuned Saka-14B model, which generates culturally appropriate answers evaluated with BERTScore.

3.1 Data Processing and Knowledge Base Construction

3.1.1 Input Data Preparation

The knowledge base comprises 1,000 Arabic mental health question–answer pairs in JSON format. Of these, 500 were provided by competition organizers, while 500 were synthetically generated using large language models (LLMs) and subsequently reviewed by human annotators.

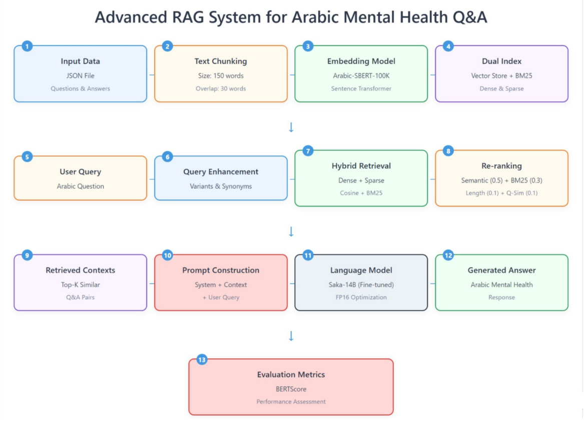


Figure 1: Advanced RAG System Architecture for Arabic Mental Health Q&A. The system processes input data through 13 main stages.

Human Verification Process: Three native Arabic speakers with expertise in mental health terminology reviewed all synthetic entries for correctness, clarity, and cultural appropriateness.

3.2 Response Classification for Semantic Categorization

A multi-label classification module categorizes responses into Information, Direct Guidance, or Emotional Support.

3.2.1 Transformer Based Models

Finetuned three transformer based models:

- **AraBERTv2**: Optimized for Modern Standard Arabic, effective for formal health content.
- **MARBERTv2**: Tuned for dialectal Arabic, capturing colloquial expressions common in mental health queries.
- **RoBERTa (English)**: Included as a cross lingual baseline to evaluate adaptation to Arabic after finetuning, quantifying the value of Arabic specific pretraining.

3.2.2 Integration with RAG Pipeline

Predicted categories from AraBERTv2 (the best performer) are stored as query metadata.used in retrieval by prioritizing contexts matching the desired strategy.

3.3 Embedding and Vector Storage

Employing Arabic-SBERT-100K for embedding generation due to its superior semantic representation of Arabic mental health language. Compared to multilingual alternatives, it better captures idiomatic expressions and domain specific terms. The 768 dimensional embeddings require more memory but significantly improve retrieval quality, justifying the storage overhead for this domain.

3.4 Query Processing and Enhancement

3.4.1 User Query Analysis

Arabic queries are normalized (e.g., removing diacritics, standardizing alef forms) while preserving meaning to prevent retrieval mismatches.

3.4.2 Query Enhancement Mechanism

To handle dialect variation, colloquial terms are expanded to their formal equivalents.

3.5 Information Retrieval and Re-ranking

3.5.1 Similarity Search

Enhanced queries are embedded using Arabic-SBERT-100K, ensuring retrieval consistency.

3.5.2 Multi-factor Re-ranking Algorithm

Contexts are ranked using:

- Semantic similarity (0.4)
- BM25 score (0.2)
- Text length (0.2)
- Question similarity (0.2)

Weights were determined through empirical tuning on a validation set, achieving the highest BERTScore F1.

3.5.3 Cultural Sensitivity Filtering

Retrieved contexts containing culturally risky recommendations (e.g., suggesting alcohol consumption as a coping method) are deprioritized or replaced with culturally acceptable alternatives (e.g., meditation, prayer, or physical exercise).

3.6 Response Generation

3.6.1 Model Choice

We fine-tuned Saka-14B for Arabic mental health support using QLoRA with parameter-efficient tuning. A dataset of user questions, assistant answers, and expert ratings was reformatted into a text-to-text causal LM style with a system prompt defining the assistant's role.

The model was loaded in 4-bit NF4 quantization with BitsAndBytes for memory efficiency, then adapted using LoRA on attention and feed-forward layers (r=8, =16, dropout=0.05). Training used

Hugging Face's Trainer with AdamW, cosine learning rate scheduling (2e-5), gradient checkpointing, FP16, and early stopping.

3.6.2 Prompt Construction

Prompts combine:

- Domain specific instructions (mental health scope)
- Top 5 retrieved contexts
- Original user query
- Cultural constraints (e.g., avoid contradicting Islamic practices)

4 Results and Evaluation

Evaluating the system across the three AraHealthQA 2025 subtasks: question categorization, answer strategy classification, and generative question answering (QA).

4.1 Evaluation Framework

4.1.1 Metrics

Using BERTScore for semantic similarity and domain specific human evaluations for appropriateness. BERTScore is an evaluation metric for text generation tasks (like machine translation, summarization, or dialogue systems) that measures semantic similarity between a candidate text and a reference text.

Instead of relying on exact word matches (like BLEU or ROUGE), BERTScore uses contextual embeddings from pretrained transformer models (e.g., BERT, RoBERTa) to capture meaning.

4.2 Question Categorization Results

Table 1 shows the class distribution for question categories in the training data.

Table 1: Distribution of Question Categories

Category	Count	Percentage
Treatment	240	24.0%
Diagnosis	210	21.0%
Healthy Lifestyle	190	19.0%
Epidemiology	85	8.5%
Other	275	27.5%

The best performing model was the MARBERTv2 ensemble, achieving:

- Weighted F1: 0.832
- Jaccard Score: 0.681
- Macro F1: 0.698

Table 2 shows the question scores distribution.

Table 2: Question Categorization Results

Model	Weighted F1
aubmindlab/bert-base-arabertv2	0.81
UBC-NLP/MARBERTv2	0.83
FacebookAI/roberta-base	0.77

4.3 Answer Categorization Results

Table 3 shows the answer strategy distribution.

Table 3: Answer Strategy Distribution

Strategy	Count	Percentage
Information	227	45.4%
Direct Guidance	173	34.6%
Emotional Support	37	7.4%

The best performing model (AraBERTv2) achieved:

- Weighted F1: 0.8289
- Jaccard Score: 0.7667

Challenge: Emotional Support detection was the most difficult due to subtle cue recognition in Arabic, where supportive intent is often expressed indirectly.

Table 4: Performance comparison of models on the Arabic text classification task.

Model	Weighted F1	Jaccard	Direct Guidance F1
AraBERTv2	0.8289	0.7667	0.786
MARBERTv2	0.8083	0.6800	0.730
RoBERTa-base	0.7710	0.6333	0.701

4.4 Question Answering Results

Comparing three configurations, as shown in the figure 2.

Statistical Significance

performed a paired bootstrap significance test ($n = 1,000$ samples) comparing Finetuned Saka-14B to the strongest baseline (Qwen14). Results showed that the improvement in BERT-F1 was statistically significant.

The finetuned Saka-14B (RAG) model achieved the highest semantic alignment with gold answers,

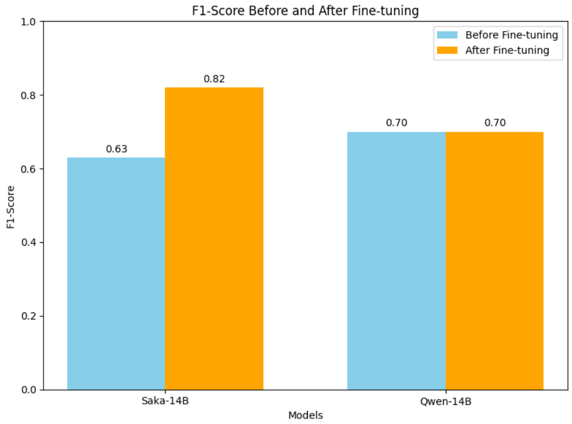


Figure 2: Bert-F1 scores of the 4 models, A comparison between Finetuning and Baseline models

benefiting from retrieval guided context selection and cultural adaptation. The Baseline Qwen14 captured some semantic similarity but often failed in cultural context handling, while Baseline Saka-14B provided verbose but less targeted responses.

Trade off Observation: Verbosity in Saka-14B outputs was partially controlled by the prompt design and top 5 retrieval limit; increasing retrieval scope tended to increase detail but sometimes reduced focus.

5 Conclusion and Future work

This study presented a multi-task Arabic mental health question answering framework that integrates question categorization, answer strategy classification, and culturally sensitive Retrieval Augmented Generation. The system addresses the linguistic complexity and cultural considerations of Arabic mental health discourse by combining semantic chunking, query enhancement, hybrid retrieval, and a finetuned Saka-14B model aligned with cultural norms.

Experimental evaluation demonstrated that the framework achieved weighted F1-scores of 0.742 for question categorization and 0.718 for answer strategy classification, alongside a BERTScore F1 of 0.821 for generative answering—up to 15% higher than retrieval only baselines. Expert evaluations confirmed that the integration of cultural filtering improved trustworthiness and contextual relevance.

Despite these promising results, limitations remain. The dataset size is small relative to comparable English language resources, which restricts the model's generalizability. Dialectal imbalance, particularly the predominance of Egyptian Arabic, impacted performance in Gulf and Levantine varieties. Furthermore, the current system lacks automated mechanisms for handling high risk cases such as suicidal ideation, and no standardized protocol for measuring cultural appropriateness has yet been established. Looking ahead, the next stage of development will focus on both scaling and refining AraHealthQA 2025. Expanding the dataset in collaboration with Arabic speaking mental health organizations is a priority, ensuring a richer variety of topics and balanced coverage of regional dialects. This expansion will provide a stronger foundation for training models that can generalize across the linguistic and cultural diversity of the Arabic speaking world.

References

[1] Yu Jin, Jiayi Liu, Pan Li, Baosen Wang, Yangxinyu Yan, Huilin Zhang, Chenhao Ni, Jing Wang, Yi Li, Yajun Bu, and Yuanyuan Wang. (2025, May). The Applications of Large Language Models in Mental Health: Scoping Review. *Journal of Medical Internet Research*, vol. 27, p. e69284. JMIR Publications.

[2] Zhijun Guo, Alvina Lai, Johan Hilmar Thygesen, Joseph Farrington, Thomas Keen, and Kezhi Li. (2024, Oct.). Large Language Models for Mental Health Applications: Systematic Review. *JMIR Mental Health*, vol. 11, p. e57400. JMIR Publications.

[3] He Hu, Yucheng Zhou, Juzheng Si, Qianning Wang, Hengheng Zhang, Fuji Ren, Fei Ma, and Laizhong Cui. (2025, May). Beyond Empathy: Integrating Diagnostic and Therapeutic Reasoning with Large Language Models for Mental Health Counseling. *arXiv preprint arXiv:2505.15715*.

[4] Lei Yin, Yuxuan Zhang, Xiaoyu Wang, and Jing Chen. (2025). Evaluating LLMs and RAG Pipelines for Mental Health App Recommendations. *AIMS Applied Computing and Informatics*, vol. 2025, no. 1, p. 10. AIMS Press.

[5] Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. (2024). MentalQA: An Annotated Arabic Corpus for Questions and Answers of Mental Healthcare. *IEEE Access*, vol. 12, pp. 101155–101165. doi: 10.1109/ACCESS.2024.3430068.

[6] Noureldin Zahran, Aya Elsayed Fouda, Radwa Jamal Hanafy, and Mohammed Elsayed Fouda. (2025, Jan.). A Comprehensive Evaluation of Large Language Models on Mental Illnesses in Arabic Context. *arXiv preprint arXiv:2501.06859*.

[7] Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaymae Abouzahir, Mohammad Abu-Daoud, Abdulrahman Alasmari, Waseem Al-Eisawi,

Rawan Al-Monef, Ali Alqahtani, Lina Ayash, Nizar Habash, and Lina Kharouf. (2025). AraHealthQA 2025: The First Shared Task on Arabic Health Question Answering. *arXiv preprint* arXiv:2508.20047, v2.

[8] Hassan Alhuzali and Ashwag Alasmari. (2025, Apr.). Pre-Trained Language Models for Mental Health: An Empirical Study on Arabic Q&A Classification. *Healthcare*, vol. 13, no. 9, p. 985. MDPI.

AraMinds at AraHealthQA 2025: A Retrieval-Augmented Generation System for Fine-Grained Classification and Answer Generation of Arabic Mental Health Q&A

Mohamed Zaytoon, Ahmed Salem, Ahmed Sakr, Hossam Elkordi
Department of Computer and Systems Engineering
Alexandria University, Egypt
{mohamed.zaytoon24,es-AhmedMahmod2022,
es-ahmedsakr20,es-hossam.elkordi2018}@alexu.edu.eg

Abstract

We present a mental health support system for Arabic that can classify both patient questions and doctor answers, and generate answers for new questions. The classification model organizes the input text to understand better the intent of the user and the response style, while the generation model produces accurate and empathetic responses. In evaluations, our system ranked 3rd in answer classification and 4th in answer generation, with only a small margin from the top-ranked systems. These results highlight the effectiveness of multi-label classification and RAG for improving access to mental health information and support in Arabic.

1 Introduction

Mental health and human psychology have been studied and practiced as separate fields of medicine for centuries (Grob, 1998), yet, studies show that the general population in the MENA region still refrains from seeking medical help when it comes to mental health-related problems, mainly due to social considerations (Nazmy, 2025), leading to a significant degradation in the public mental health status(Altobaishat et al., 2025).

After the rise of large language models (LLMs) and Agentic Artificial intelligence (AI) applications (Minaee et al., 2024; Plaat et al., 2025), especially for complex reasoning and questions answering (QA) tasks, many researches have explored the use of advanced language models to provide not only assistance for medical professionals (Nazi and Peng, 2024), but also as an alternative means of delivering mental health care (Guo et al., 2024), due to their ability to provide human like responses and interactions (Zaki and Hassan, 2023; Zahran et al., 2025), offering scalable, accessible, private, and stigma-free pathways for psychological support.

Besides the aforementioned cultural barriers, providing automated mental health support for Arabic speakers faces other challenges, including:

- **Higher Accuracy Standard:** the medical field -especially psychology- has a very low tolerance of error, as opposed to other AI applications, where in some cases, accuracy could be traded off for speed or power efficiency (Han et al., 2015), the cost of error in medical applications could lead to unquantifiable losses, causing -in the worst cases- human fatalities (Topol, 2019).

- **Data Scarcity:** this challenge is two-fold: 1) Arabic datasets are scarce and generally have lower quality annotations in general, 2) Datasets for mental health-related problems (Alhuzali et al., 2024) are not as abundant as other health-related datasets (**?**Alasmari, 2025).

- **Patient Confidentiality:** unlike other medical disciplines, obscuring patient identity is more challenging, as personal information, such as background and upbringing circumstances, has to be included in every case.

- **Linguistic Complexity:** Apart from data problems, Arabic is morphologically rich and includes many dialects with a high level of diversity (Habash, 2010), which leads to a wide performance gap of language models between Arabic and other languages.

Our contribution in this task could be summarized as:

- We developed a classification system to label questions and answers into multiple fine-grained categories.

- We integrate additional external knowledge from Arabic medical platforms to develop

Proceedings of The Third Arabic Natural Language Processing Conference, pages 198–203
November 8-9, 2025 ©2025 Association for Computational Linguistics

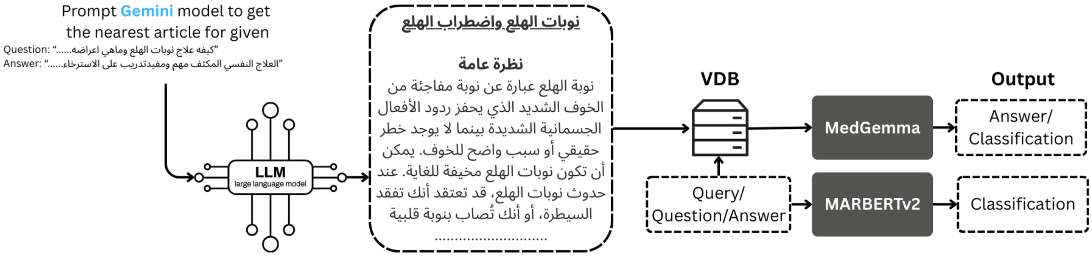

Prompt Gemini model to get
the nearest article for given
Question: "كيفه علاج نوبات الهلع وماهي اعراضه......"
Answer: "العلاج النفسي المكثف مهم ومفيدتدريب على الاسترخاء......"
LLM
large language model
نوبات الهلع واضطراب الهلع
نظرة عامة
نوبة الهلع عبارة عن نوبة مفاجئة من
الخوف الشديد الذي يحفز ردود الأفعال
الجسمانية الشديدة بينما لا يوجد خطر
حقيقي أو سبب واضح للخوف. يمكن
أن تكون نوبات الهلع مخيفة للغاية. عند
حدوث نوبات الهلع، قد تعتقد أنك تفقد
السيطرة، أو أنك تُصاب بنوبة قلبية
....
VDB
Query/
Question/Answer
MedGemma
MARBERTv2
Output
Answer/
Classification
Classification

Figure 1: General overview of our system starting from the document collection phase, the RAG system for all three subtasks, and multi-label classification, and question answering.

a simple yet effective retrieval-augmented generation (RAG) system tailored to Arabic mental health Q&A, improving overall classification accuracy and reducing hallucination in answers.

- We achieved 0.56[1] f1 score on the first track, 0.76 on the second one, and 0.663 on the BERTScore (Zhang et al., 2020) metric.

2 Background

The challenge (Alhuzali et al., 2025) is divided into three main sub-challenges. The first two are **multilabel classification** for human questions and their answers, respectively, with seven different classes for questions and three for answers. The third sub-challenge is **answer generation** to provide mental health assistance to patients by generating contextually appropriate and medically accurate responses to user queries.

A lot of work has been put into data collection for mental health, from different sources such as social media platforms, for example *Reddit* (Cohan et al., 2018; Di Cara et al., 2023). This data is then manually annotated to identify different mental health-related conditions, such as depression detection (Han et al., 2022), anxiety, bipolar disorder, and suicidal intent (Ji et al., 2022).

Recent work explored the capabilities of LLMs in the mental health domain for different tasks. First, classification tasks to detect different mental health conditions (Racha et al., 2025) and their causes (Yang et al., 2023). Second, user question answering, to respond to different user queries, either informative or for advice seeking. Third, chatbots offering their users a safe space for relief and conversation (Shan et al., 2022) or chatbots that help doctors in the application and monitoring of different treatments, including cognitive behavioral therapy (CBT) (Farzan et al., 2025), and self-attachment technique (SAT) (Elahimanesh et al., 2023).

The use of external knowledge in Retrieval Augmented Generation (RAG) systems to enhance the factuality and robustness of LLMs, especially in the medical domain, has been explored, for example (Vladika and Matthes, 2024) retrieved from a large corpus that included diverse topics such as *dietary supplements, heart and lungs, reproductive health, cancer, and mental health.* This demonstrated that retrieval strategies prioritizing fewer, more recent, and highly cited sources—especially at the sentence level—significantly improved answer quality in health question answering tasks.

Recently, many researchers have taken an interest in the mental health domain in the Arabic language. starting with simpler classification tasks, such as depression detection (Maghraby and Ali, 2022; Hassib et al., 2022). Others worked on more advanced tasks such as (Zahran et al., 2025).

3 System Overview

In this section, we present our system setup in detail. First, we go through the data collection process, then we describe our RAG pipeline, used models, and embedding vectorestore. Finally, we go through how we built our multi-label classifier for the question and answer classification tracks.

3.1 Building Knowledge Base

To effectively build our knowledge base, we followed a two-step process. First, we used the Gemini API[2] to collect articles related to each question from the training set. Then, to reduce the size of the knowledge base and guarantee smoother retrieval of relevant information, we fed articles with

[1] due to the limited number of submissions, this score didn't show on the leaderboard, the metrics were computed in the post-evaluation phase.

[2] https://gemini.google.com

Task	RAG	Pretraining	F1-score
Question Classification	✓	✗	0.490
	✗	✓	**0.558***
	✗	✗	0.549*
Answer Classification	✓	✗	**0.760**
	✗	✓	0.735*
	✗	✗	0.733*

Table 1: Macro-F1 scores on the test set using the two approaches we developed, RAG and multi-label classification with and without continuous pretraining. (*) denotes experiments done during the post-evaluation phase, and were not submitted to the leaderboard due to the limited number of submissions.

high relevance to the same question to Gemini to compress them into one article. We then used the same knowledge base for both the evaluation and test phases, without any updates from the test set.

3.2 Retrieval Augmented Generation Pipeline

After the data collection phase, we embedded those related articles using `BAAI/bge-m3` (Chen et al., 2024) for its *multi-functional*, *multi-granular*, and especially *multi-lingual* capabilities. Those embeddings are then stored in a `chroma-db`[3] vectorstore. For every sample, we retrieved the top articles related to it and used them as additional context to generate answers for different classification and generation tasks using `google/medgemma-4b-it` (Sellergren et al., 2025).

Model	RAG	Validatoin	Test
Qwen2.5 7b	✗	0.608	–
Llama3.2 3b	✗	0.597	–
Phi-mini 4b	✗	0.606	–
Gemma3-4b	✗	0.619	–
MedGemma3-4b	✗	0.620	0.632
MedGemma3-4b	✓	**0.630**	**0.663**

Table 2: BertScore results on the validation and test sets for the answer generation sub-task.

3.3 Multi-label Classification for Question and Answer

For the multi-label classification tasks, we relied on `MARBERTv2` (Abdul-Mageed et al., 2021) model. To improve the model's understanding capabilities in the mental health domain, we applied masked language modeling (MLM) pretraining (Devlin et al., 2019) using different Arabic mental health books such as DSM-5 (EDITION, 1980) and articles about different psychiatric and mental health conditions from the renowned *Royal College Of Psychiatrists* [4]. The pretrained model was then fine-tuned for both the question and answer multi-label classification tasks, using the *binary cross-entropy* loss–reference the loss function–.

4 Experimental Setup

4.1 Dataset

4.1.1 Shared Task Dataset - MentalQA

The given dataset includes two different tasks: multi-label classification and answer generation. The classification part is for both questions and answers, and the generation is to reply to the given question. The QA pairs were collected from *altibbi* [5] medical platform for advisory and information, then 500 samples were manually annotated (Alhuzali et al., 2024). Questions are categorized into seven classes, while answers are classified into only three.

4.1.2 Knowledge Base Dataset

The data was collected using the Gemini API, the pre-comrpression articles were mainly from Arabic medical websites, such as `islamweb` [6], `Mayo Clinic` [7], `Mind Clinic Group` [8]. The collected articles were then curated as mentioned in section 3.1. The compressed articles were then added to a vectorstore database to facilitate retrieval

For the rag system, we use `BAAI/bge-m3` model (Chen et al., 2024), and to store the data, we used chroma-db.

4.1.3 Continue Pretraining Dataset

To better ground our base model's capacity in understanding text from the mental health domain, we utilized text scraped from multiple resources, including 32 articles from the *Royal College of Psychiatrists* and a collection of 72 books, either originally written in Arabic or manually translated into Arabic by professional human translators. The combined dataset contained approximately 4.5 million tokens.

[3] https://github.com/chroma-core/chroma

[4] https://www.rcpsych.ac.uk/mental-health/translations/arabic

[5] https://altibbi.com

[6] https://islamweb.net/ar/

[7] https://www.mayoclinic.org/ar

[8] https://mindclinicgroup.com/ar

Task	Input	Ground Truth	Prediction
Question Classification	ماهو افضل دواء منوم وذاا تأثير سريع وقوي لاني اعاني من ارق ولا استطيع النوم ابدااا الرجاء الاجابه؟؟	Treatment	Diagnosis Anatomy and Physiology Epidemiology
Answer Classification	لا يجوز اخذ هذه الادوية دون استشارة الطبيب لان لها اثار جانبية كثيرة فيجب مراجعة الطبيب	Information Direct Guidance	Direct Guidance
Answer Generation	لما تصير مشكله تافهه او قويه مع اشخاص اصير افكر فيها فجأه واخلق سيناريوهات وحوار كله هواش لدرجة بالواقع علاقتي تأثرت بهالاشخاص حاولت اتجاهل الافكار بس فجأه عقلي يفكر فيها	مهم العلاج النفسي والتدريب على التركيز والاسترخاء	يبدو أنك تعاني من أفكار سلبية ومشاعر قلق بشأن علاقاتك مع الآخرين، مما يؤثر على حياتك اليومية. هذه المشاعر قد تكون ناتجة عن سيناريوهات تخيلية أو حوارات افتراضية مليئة بالجدال.**إليك بعض الاقتراحات التي قد تساعدك:*** **التحدث مع شخص تثق به:** مشاركة مشاعرك ....

Table 3: Some fauiler cases for our system on the three sub tasks.

4.2 Training Details

To continue the pretraining of the base MARBERTV2 model, we used the AdamW optimizer, with a learning rate 2e-5, 0.01 weight decay, a cosine annealing learning rate scheduler, and a batch size of 32. We trained the model for 20 epochs. For training the Multi-Label classification models, in both sub-tasks 1,2, we used the same optimizer and learning rate, with a linear scheduler, and a batch size of 4. Both models were trained for 10 epochs. All training was done on a single NVIDIA RTX-3090 GPU.

5 Results

5.1 Classification Tasks Results

For the classification task, we tested the RAG approach to label both questions and answers separately. Then, we compared this approach with our multi-label classification. Also, we assessed the effect of the continue pretraining phase. We can see from table 1 that in both cases the pretraining phase improved the results by 0.009 and 0.002 on the question and answer classification tasks, respectively. Also, the RAG approach achieved better results in the answer classification task by 0.025, but came short by 0.068 in the question classification task.

5.2 Generation Tasks Results

To choose the best model, we run some initial tests on the answer generation task using various open-source LLMs. After picking the highest scoring model, we used it for the remaining experiments in our RAG system. The results of this experiment are shown in table 2.

5.3 Analysis

Table 3 presents failure cases from our system. In the **Question Classification**, the model misclassifies a request for a sleeping medication as a query about *Diagnosis*, *Anatomy and Physiology*, and *Epidemiology*, indicating a misunderstanding of user intent by over-focusing on the symptom keyword أرق (insomnia). For **Answer Classification**, the system correctly identifies *Direct Guidance* but misses the *Information* label, showing challenges in capturing all nuances. In **Answer Generation**, responses are verbose and generic, such as suggesting التحدث مع شخص تثق به (Talk to someone you trust), instead of specific, actionable advice, underscoring a preference for supportive text over professional recommendations.

6 Conclusion

This paper investigated our work in the first track of the AraHealthQA 2025 shared task for mental health question and answer multi-label classification and answer generation. We collected a large corpus of Arabic mental health-related books and articles and used them to continue pretraining our base encoder. Then, we fine-tuned this model on the classification tasks. We also benefited from the provided questions and collected articles from the internet using an agentic search tool. These articles are then used in a retrieval-augmented generation system for all three sub-tasks. Our system achieved 0.558 and 0.760 F1-score on question and answer multi-label classification tasks, respectively, while achieving 0.663 BertScore on the answer generation task.

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.

Ashwag Alasmari. 2025. A scoping review of arabic natural language processing for mental health. In *Healthcare*, volume 13-9, page 963. MDPI.

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*, 12:101155–101165.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash, and Leen Kharouf. 2025. Arahealthqa 2025 shared task description paper. In *Proceedings of Arabic-NLP 2025*.

Obieda Altobaishat, Mohamed Abouzid, Deemah Omari, Walid Sange, Ahmad K Al-Zoubi, Abdallah Bani-Salameh, and Yazan A Al-Ajlouni. 2025. Examining the burden of mental disorders in jordan: an ecological study over three decades. *BMC psychiatry*, 25(1):218.

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. *Preprint*, arXiv:2402.03216.

Arman Cohan, Bart Desmet, Andrew Yates, Luca Soldaini, Sean MacAvaney, and Nazli Goharian. 2018. Smhd: a large-scale resource for exploring online language usage for multiple mental health conditions. *arXiv preprint arXiv:1806.05258*.

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.

Nina H Di Cara, Valerio Maggio, Oliver SP Davis, and Claire MA Haworth. 2023. Methodologies for monitoring mental health on twitter: systematic review. *Journal of Medical Internet Research*, 25:e42734.

FIFTH EDITION. 1980. Diagnostic and statistical manual of mental disorders. *American Psychiatric Association, Washington, DC*, pages 205–224.

Sina Elahimanesh, Shayan Salehi, Sara Zahedi Movahed, Lisa Alazraki, Ruoyu Hu, and Abbas Edalat. 2023. From words and exercises to wellness: Farsi chatbot for self-attachment technique. *arXiv preprint arXiv:2310.09362*.

Maryam Farzan, Hamid Ebrahimi, Maryam Pourali, and Fatemeh Sabeti. 2025. Artificial intelligence-powered cognitive behavioral therapy chatbots, a systematic review. *Iranian journal of psychiatry*, 20(1):102.

Gerald N Grob. 1998. A history of psychiatry: From the era of the asylum to the age of prozac. *Bulletin of the History of Medicine*, 72(1):153–155.

Zhijun Guo, Alvina Lai, Johan H Thygesen, Joseph Farrington, Thomas Keen, Kezhi Li, and 1 others. 2024. Large language models for mental health applications: systematic review. *JMIR mental health*, 11(1):e57400.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Song Han, Huizi Mao, and William J. Dally. 2015. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *arXiv: Computer Vision and Pattern Recognition*.

Sooji Han, Rui Mao, and Erik Cambria. 2022. Hierarchical attention network for explainable depression detection on Twitter aided by metaphor concept mappings. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 94–104, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Mariam Hassib, Nancy Hossam, Jolie Sameh, and Marwan Torki. 2022. Aradepsu: Detecting depression and suicidal ideation in arabic tweets using transformers. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 302–311.

Shaoxiong Ji, Xue Li, Zi Huang, and Erik Cambria. 2022. Suicidal ideation and mental disorder detection with attentive relation networks. *Neural Comput. Appl.*, 34(13):10309–10319.

Ashwag Maghraby and Hosnia Ali. 2022. Modern standard arabic mood changing and depression dataset. *Data in Brief*, 41:107999.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Zabir Al Nazi and Wei Peng. 2024. Large language models in healthcare and medical domain: A review. In *Informatics*, volume 11, page 57. MDPI.

Sarah Aly Nazmy. 2025. *Cultural Factors and Mental Health Help-Seeking Behaviors Among Middle Eastern/North African Adults in the United States*. Ph.D. thesis, Alliant International University.

Aske Plaat, Max van Duijn, Niki van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. 2025. Agentic large language models, a survey. *arXiv preprint arXiv:2503.23037*.

Suraj Racha, Prashant Joshi, Anshika Raman, Nikita Jangid, Mridul Sharma, Ganesh Ramakrishnan, and Nirmal Punjabi. 2025. Mhqa: A diverse, knowledge intensive mental health question answering challenge for language models. *arXiv preprint arXiv:2502.15418*.

Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, Shawn Xu, Fayaz Jamil, Cían Hughes, Charles Lau, and 1 others. 2025. Medgemma technical report. *arXiv preprint arXiv:2507.05201*.

Yong Shan, Jinchao Zhang, Zekang Li, Yang Feng, and Jie Zhou. 2022. Mental health assessment for the chatbots. *arXiv preprint arXiv:2201.05382*.

Eric J Topol. 2019. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56.

Juraj Vladika and Florian Matthes. 2024. Improving health question answering with reliable and time-aware evidence retrieval. *arXiv preprint arXiv:2404.08359*.

Kailai Yang, Tianlin Zhang, Ziyan Kuang, Qianqian Xie, and Sophia Ananiadou. 2023. Mentalllama: Interpretable mental health analysis on social media with large language models. *arXiv preprint arXiv:2309.13567*.

Mohamed Zahran and 1 others. 2025. A comprehensive evaluation of large language models on mental illnesses in arabic context. *arXiv preprint arXiv:2501.06859*.

A Zaki and R Hassan. 2023. Optimizing large language models for arabic healthcare communication: A focus on patient-centered nlp applications. *Multimodal Technologies and Interaction*, 8(11):157.

Tianyi Zhang, Kishore, Wu, Q. Weinberger, and Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Fahmni at AraHealthQA Track 1: Multi-Agent Retrieval-Augmented Generation and Multi-Label Classification for Arabic Mental Health Q&A

Caroline Sabty
German International University
caroline.sabty@giu.edu.eg

Mohamad Rasmy
MBZUAI
m.rasmy@mbzuai.ac.ae

Mohamed Eyad Badran
Gameball Company
mohamed.eyad2612@gmail.com

Nourhan Sakr
American University in Cairo
n.sakr@columbia.edu

Alia El Bolock
American University in Cairo
alia.elbolock@aucegypt.edu

Abstract

We present FAHMNI, a unified system for Arabic mental-health question answering developed for the AraHealthQA 2025 MentalQA Shared Task (Track 1). FAHMNI evaluates large language models (LLMs) on all subtasks: (1) multi-label classification of question types and (2) answer strategies, and (3) grounded answer generation. For Subtasks 1–2, we systematically compare Arabic-capable LLM families (Qwen3, SILMA) under zero-shot and few-shot prompting, few-shot learning with a frozen backbone, parameter-efficient fine-tuning (PEFT), and instruction tuning. To support Subtask 3, we implement a multi-agent, retrieval-augmented generation pipeline that routes queries between curated domain sources and controlled web search; an answer-style controller predicts the required strategy (Information, Direct Guidance, Emotional Support) and conditions the generator accordingly. Our best LLM configurations reach 0.507/0.404 (weighted-F1/Jaccard) on Subtask 1 with Qwen3+PEFT and 0.750/0.600 on Subtask 2 with SILMA+PEFT, while a strong fine-tuned MARBERT baseline remains competitive at 0.541/0.494 (Subtask 1) and 0.805/0.727 (Subtask 2). For Subtask 3, our multi-agent RAG system with SILMA attains an 0.652 BERTScore F1 and yields a 0.06 hallucination rate under our manual audit. These findings highlight both the viability and current limits of Arabic-capable LLMs for mental-health QA, and they motivate grounded, style-aware generation as a practical path for safe deployment.

1 Introduction

Despite the growing global awareness of mental health needs, Arabic remains severely underrepresented in mental health NLP resources. Existing work on Arabic mental health question answering (QA) is limited in both scale and task coverage, hindering the development of reliable digital support tools, e.g. triage, education, guided self-help, for Arabic speakers. Complementary efforts on mental health text classification, such as cognitive distortions detection with data augmentation (Rasmy et al., 2024), highlight the importance of tailored augmentation for improving robustness in this sensitive domain. The AraHealthQA 2025 shared task (Track 1) (Alhuzali et al., 2025) addresses this gap by introducing Arabic *mental-health QA* across three subtasks: (1) multi-label classification of question types; (2) multi-label classification of answer strategies; and (3) answer generation (Alhuzali et al., 2024). To tackle all three subtasks, we develop FAHMNI, a unified system for Arabic mental-health QA. Our system leverages two modern Arabic-capable LLM families: **Qwen3** and **SILMA** (SILMA9BInstruct, 2024) **/ Kashif** family (SILMA-AI, 2025), motivated by the strength of their predecessors in multilingual transfer for Arabic health retrieval and QA on the Massive Text Embedding Benchmark (MTEB) (Enevoldsen et al., 2025) and their competitive Arabic benchmarks including Arabic RAG-style QA on the Arabic Broad Leaderboard (Ouda, 2025).

Our approach for Subtasks 1 and 2 compares zero-shot prompting, few-shot prompting, and few-shot learning under both frozen-backbone, parameter-efficient fine-tuning (PEFT), and instruction tuning regimes. For Subtask 3, we design a multi-agent, retrieval-augmented answer generation system that dynamically routes queries, integrates curated domain resources, and invokes open-web retrieval when coverage is insufficient. We summarize our contributions as follows:

1. **Comprehensive evaluation** of state-of-the-art Arabic-capable LLMs on all three AraHealthQA subtasks, spanning prompting and fine-tuning strategies.
2. **A novel multi-agent, retrieval-augmented architecture** that moves beyond prior classification-only evaluations and offers

Proceedings of The Third Arabic Natural Language Processing Conference, pages 204–212
November 8-9, 2025 ©2025 Association for Computational Linguistics

grounded answer generation.

3. **Reproducible resources** including code, configurations, and prompts to support future Arabic mental-health QA research.

Our system achieves competitive results across all three subtasks: For Subtask 1, we use Qwen3 under PEFT (weighted-F1 = 0.51; Jaccard = 0.4) and for Subtask 2, we choose SILMA under PEFT (weighted-F1 = 0.75; Jaccard = 0.6). Finally, for Subtask 3, the SILMA Kashif model reaches a BERTScore of 0.652. In practice, we encountered three recurring challenges: label overlap across clinically adjacent categories, dialectal and terminology variation, and limited data availability due to the small training split. Our code is available at `https://github.com/MHRasmy/AraHealthQA-2025-Track-1`.

2 Background

The shared task uses the MentalQA corpus of Arabic patient–doctor Q&A pairs annotated for seven *question types* (Diagnosis, Treatment, Anatomy & Physiology, Epidemiology, Healthy Lifestyle, Provider Choice, Other) and three *answer strategies* (Information, Direct Guidance, Emotional Support) (Alhuzali et al., 2024). The annotation study reports substantial reliability (Fleiss' $\kappa = 0.61$ for question types; $\kappa = 0.96$ for answer strategies). Track 1 releases 500 Q&A posts with splits of 350 (train_dev) and 150 (test). Official metrics are weighted F1 and Jaccard for Subtasks 1–2 (multi-label classification), and BERTScore for Subtask 3 (grounded answer generation conditioned on classifications). For readers unfamiliar with MentalQA-style Q&A posts, we include illustrative Arabic examples in Appendix A.

Previous benchmarks (Alhuzali and Alasmari, 2025) compared classical SVM features, frozen PLM encoders, fine-tuned Arabic PLMs (e.g., AraBERT, CAMeLBERT, MARBERT), and GPT-3.5/4-based prompting. The fine-tuned MARBERT showed strongest classification performance, with few-shot prompting outperforming zero-shot. We adopt this model as a well-established baseline and extend the line of work by evaluating newer Arabic-capable LLMs (Qwen3, SILMA) under zero-shot, few-shot prompting, few-shot learning, fine-tuning, and instruction tuning regimes for Subtasks 1–2, and by by operationalizing grounded answer generation for Subtask 3 via a multi-agent, retrieval-augmented design.

3 System Overview

In this work, we introduce FAHMNI, a single, modular architecture that couples classification and grounded generation, thereby addressing all three AraHealthQA Track 1 subtasks.

Subtasks 1–2 (multi-label classification). For Tasks 1 and 2, we evaluate five approaches with the Arabic-capable LLM families **Qwen3** and **SILMA**: *zero-shot prompting*, *few-shot prompting*, *few-shot learning*, *PEFT*, and *instruction tuning*. In zero-shot prompting, models receive only label definitions; few-shot prompting augments this with compact, label-balanced exemplars. To move beyond prompting without overfitting in a small-data setting, we train a shallow classification head over frozen LLM representations ("few-shot learning"). Finally, we perform instruction tuning in zero- and few-shot settings. This progression lets us quantify how much the task benefits from parametric specialization versus prompt conditioning under multi-label imbalance and clinically adjacent categories (e.g., Diagnosis vs. Treatment).

Subtask 3 (grounded answer generation via RAG). Given the sensitivity of mental-health counseling, responses should be *grounded*, factual, and style-appropriate. We, therefore, adopt retrieval-augmented generation (RAG) for Task 3, based on evidence that RAG improves faithfulness and reduces hallucinations on knowledge-intensive tasks (Lewis et al., 2020; Ayala and Bechard, 2024). Our pipeline (Fig. 1) is organized around a *decision agent*, which first inspects the query along with available candidate passages retrieved from the local knowledge base, then uses few-shot prompting (details in Appendix B) to select between a static, curated knowledge base and a dynamic web retrieval path.

(a) Static domain-specific retrieval. For well-scoped questions, the system consults a curated local knowledge base assembled from canonical references: DSM-5-TR (Association, 2022) for *Diagnosis*, OpenStax Anatomy & Physiology (Betts et al., 2024) for *Anatomy & Physiology*, CDC's Principles of Epidemiology (Edition, 2006) for *Epidemiology*, and MedlinePlus articles for *Provider Choice* and general guidance. Retrieved passages are retrieved by similarity search (Qwen3-Embedding, 4B variant) and provided as grounding for the answer. Static retrieval yields high-precision responses but is limited by coverage gaps (e.g.,

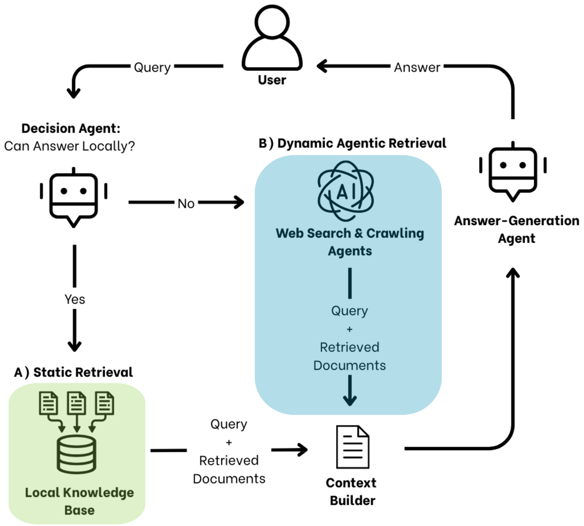


Figure 1: Multi-agent retrieval-augmented generation (RAG) pipeline for Subtask 3. A single LLM (SILMA or Qwen-3) serves *both* as the Decision/Generation agent: it first decides whether the query can be answered locally without retrieval; if not, it triggers retrieval, and later generates the final answer (details in Appendix A). Two retrieval paths are supported: (A) **Static Retrieval** from a curated local knowledge base for scoped domains, and (B) **Dynamic Agentic Retrieval** that launches web search and crawling agents to acquire evidence when curated coverage is insufficient. The retrieved documents are summarized by the agents and incorporated into the prompt, which is then provided to the LLM to produce a grounded response returned to the user.

Treatment and *Healthy Lifestyle* are too broad for a single canonical source).

(b) Dynamic agentic retrieval. For broader or open-ended queries, the decision agent triggers a web-based retrieval pipeline. Here, dedicated *Gemini-2.0-Flash* agents perform web search and crawling to acquire evidence from reliable sources (e.g., WHO, NIH, Mayo Clinic, CDC). Retrieved content is summarized by the agents, assembled into a context prompt, and then passed to the answering LLM (Qwen3 or SILMA), which generates the final grounded response.

4 Experimental Setup

Data splits. We follow the shared-task protocol: the training split contains 350 instances, which we partition into 300 for training and 50 for validation; the test set contains 150 instances. For few-shot classification, exemplars are chosen to cover *all* labels so the model observes at least one positive instance per class.

Hyperparameters. For fine-tuning in Tasks 1–2, we use a learning rate of 2×10^{-5}, batch size 8, and train for up to 10 epochs with early stopping on weighted F1 (validation split). These hyperparameters were chosen to match those in (Alhuzali and Alasmari, 2025) for consistency with the MARBERT baseline. We fix the random seed across all runs for reproducibility. For Task 3 generation, we set the temperature to 0 and disable sampling to obtain deterministic outputs for both model families.

Evaluation. Tasks 1–2 are evaluated with weighted F1 and the Jaccard index. Task 3 is evaluated with BERTScore (Zhang* et al., 2020).

Additional evaluation for Task 3 (RAG quality). Because mental health is a highly sensitive domain, we complemented standard metrics with domain-tailored ones to better capture answer quality and errors. Following Zhu et al. (Zhu et al., 2025), we report *Completeness* (coverage of extracted gold key points), *Hallucination* (contradictions), and *Irrelevance* (omissions). These metrics provide a granular view of factual reliability beyond BERTScore. Formal definitions and scoring details are given in Appendix C.

5 Results

5.1 Quantitative Performance

Table 1 reports official test-set results for Subtasks 1 (question-type classification) and 2 (answer-strategy classification) across baseline fine-tuning, few-shot prompting, parameter-efficient fine-tuning (PEFT), and instruction tuning.

For **Subtask 1**, the baseline fine-tuned model attains the strongest weighted F1 (**0.541**) and Jaccard (**0.494**). PEFT models follow (Qwen: F1 0.507; SILMA: F1 0.497), while few-shot prompting (Qwen) trails (F1 0.440). The instruction-tuned few-shot Qwen variant reaches F1 0.533 but a lower Jaccard 0.412, suggesting more partial label overlap than exact set matches.

For **Subtask 2**, the baseline fine-tuned model again leads (F1 **0.805**; Jaccard **0.727**). Among non-baseline settings, PEFT (SILMA) is strongest (F1 0.753; Jaccard 0.670), followed by instruction-tuned few-shot Qwen (F1 0.738; Jaccard 0.651). Empty predictions are rare and appear mainly in PEFT settings.

Task / Method	F1	Jac.	Empty
Subtask 1: Question Type Classification			
Baseline FT	**0.541**	**0.494**	0
Few-shot (Qwen)	0.440	0.453	0
PEFT (Qwen)	0.507	0.434	7
PEFT (SILMA)	0.497	0.422	7
Instr. Tuning Few-shot (Qwen)	0.533	0.412	0
Subtask 2: Answer Strategy Classification			
Baseline FT	**0.805**	**0.727**	0
Few-shot (Qwen)	0.622	0.572	0
PEFT (Qwen)	0.701	0.607	2
PEFT (SILMA)	0.753	0.670	1
Instr. Tuning Zero-shot (Qwen)	0.646	0.589	0
Instr. Tuning Few-shot (Qwen)	0.738	0.651	0

Table 1: Official test-set results for Subtasks 1 and 2. Best per subtask in bold.

5.2 Error Analysis

Table 2 shows the distribution of exact, partial, and wrong predictions. We expand here on why models make mistakes.

Subtask 1 (question types). The baseline FT has the highest partial-match rate (60.67%), which explains its strong F1 and Jaccard scores: it often identifies part of the correct set of question types, but misses others. Few-shot (Qwen) gives the highest exact rate (24.67%) but also the highest wrong rate (29.33%), meaning it sometimes predicts all labels correctly but more often misclassifies completely. PEFT variants stay competitive on partial matches but achieve fewer exact hits.

Looking at the labels, we see frequent misses on *Healthy lifestyle*, *Epidemiology*, and *Treatment*, while *Diagnosis* and *Treatment* are often added incorrectly. This indicates that the models sometimes confuse overlapping categories: for example, lifestyle-related questions are mistaken as treatment-related, and prognosis/etiology questions (epidemiology) are mistaken as diagnostic ones. The instruction-tuned few-shot Qwen reflects this tendency clearly: it achieves the highest partial rate (79.33%) but only 7.33% exact, as it often adds extra labels such as *Diagnosis* or *Treatment* while missing *Healthy lifestyle*. This increases recall but reduces exact agreement.

Subtask 2 (answer strategies). Here, the baseline FT achieves the best balance with the highest exact rate (48.67%) and the lowest wrong rate (3.33%). PEFT (Qwen) produces the most partial predictions (56.67%), often identifying one correct strategy but missing another. Across systems, the most common source of errors comes from *Information* and *Direct Guidance*: answers that mix factual knowledge with advice are difficult for models to consistently label, causing under-prediction or over-prediction of these two categories. Instruction-tuned few-shot Qwen improves over zero-shot by converting some wrong cases into partial matches, showing that in-context examples help the model separate advice from information.

Empty predictions. Empty outputs occur when all predicted scores fall below the decision threshold of 0.5. They are rare but appear mainly in PEFT runs (S1: 7 for Qwen, 7 for SILMA; S2: 2 for Qwen, 1 for SILMA). In these cases, the model is overly conservative, assigning low confidence to all categories and outputting no label.

Takeaways. Across both subtasks, the main challenges are (i) partial matches caused by overlapping categories, such as *Diagnosis* vs. *Treatment* or *Information* vs. *Direct Guidance*, and (ii) threshold-related errors that lead to either empty predictions or the addition of extra labels. These issues explain why the baseline FT remains the strongest overall: it provides more balanced predictions with higher exact matches, while instruction tuning (Subtask 1) trades exactness for broader coverage.

Task / Method	Exact %	Partial %	Wrong %
Subtask 1			
Baseline FT	22.67	60.67	16.67
Few-shot (Qwen)	**24.67**	46.00	29.33
PEFT (Qwen)	16.67	58.00	25.33
PEFT (SILMA)	16.67	57.33	26.00
Instr. Tuning Few-shot (Qwen)	7.33	**79.33**	**13.33**
Subtask 2			
Baseline FT	**48.67**	48.00	**3.33**
Few-shot (Qwen)	39.33	37.33	23.33
PEFT (Qwen)	33.33	**56.67**	10.00
PEFT (SILMA)	40.67	52.67	6.67
Instr. Tuning Zero-shot (Qwen)	36.67	45.33	18.00
Instr. Tuning Few-shot (Qwen)	38.67	53.33	8.00

Table 2: Error distribution for Subtasks 1 and 2. Best per column and subtask in bold.

Model	SILMA	Qwen
BERTScore ↑	**0.652**	0.645
Completeness ↑	0.567	**0.6**
Hallucination ↓	0.06	**0.04**
Irrelevance ↓	0.373	**0.36**

Table 3: Task 3 (RAG answer generation) results on the MentalQA test set. We report BERTScore F1, Completeness, Hallucination, and Irrelevance.

5.3 Task 3: Answer Generation (RAG)

Table 3 summarizes results for SILMA and Qwen-3. Both models perform similarly overall. SILMA attains a slightly higher BERTScore (0.652 vs. 0.645), while Qwen-3 achieves higher Completeness (0.600 vs. 0.567) and lower Hallucination (0.04 vs. 0.06) and Irrelevance (0.36 vs. 0.373). Qualitative best and worst examples for each model are provided in Appendix D.

The uniformly low Hallucination rates ($\leq$0.06) indicate that generated answers rarely contain content that *contradicts* the gold key points, suggesting that the RAG pipeline effectively constrains factual errors. At the same time, completeness around 0.57–0.60 shows that only about three-fifths of the gold key information is covered, leaving a substantial fraction of gold content unaddressed (Irrelevance 0.36–0.373). This explains the moderate BERTScore values ($\approx$0.65): limited key-point overlap and the inclusion of additional retrieved details (which are non-contradictory but not present in the references) dilute semantic alignment with the gold answers, lowering BERTScore despite the low Hallucination.

6 Conclusion

We presented FAHMNI, a unified system for Arabic mental-health question answering that combines multi-label classification (question types and answer strategies) with a retrieval-augmented, multi-agent generator. On Subtasks 1–2, classic Arabic PLMs remain a strong baseline: fine-tuned MARBERT delivers the best weighted F1 and Jaccard overall, while Arabic-capable LLMs (Qwen3, SILMA) with PEFT and instruction tuning are competitive under tighter compute and data budgets. On Subtask 3, both SILMA and Qwen3 yield similarly strong grounded generation with uniformly *low hallucination* rates ($\leq$ 0.06), indicating faithful adherence to evidence. At the same time, mid-range BERTScore and $\sim$0.6 completeness reveal recall gaps: answers are generally factual but do not fully cover gold key points, and extra retrieved details can dilute reference overlap.

Acknowledgments

This research was supported by Emirates NBD, Sharjah Electricity, Water & Gas Authority (SEWA), and the Technology Innovation Institute (TII), who served as the golden sponsors of the 5th Forum for Women in Research (QUWA): Together Innovating to Shape the Future at the University of Sharjah.

References

Hassan Alhuzali and Ashwag Alasmari. 2025. Pre-trained language models for mental health: An empirical study on arabic qa classification. *Healthcare*, 13(9).

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*, 12:101155–101165.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash, and Leen Kharouf. 2025. Arahealthqa 2025 shared task description paper. In *Proceedings of ArabicNLP 2025*.

American Psychiatric Association. 2022. *Diagnostic and Statistical Manual of Mental Disorders: DSM-5-TR*. American Psychiatric Association Publishing.

Orlando Ayala and Patrice Bechard. 2024. Reducing hallucination in structured outputs via retrieval-augmented generation. 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 228–238, Mexico City, Mexico. Association for Computational Linguistics.

J Gordon Betts, Kelly A Young, James A Wise, Eddie Johnson, Brandon Poe, Dean H Kruse, Oksana Korol, Jody E Johnson, Mark Womble, and Peter DeSaix. 2024. *Anatomy and physiology 2e*.

Third Edition. 2006. Principles of epidemiology.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, and 67 others. 2025. Mmteb: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Karim Ouda. 2025. ABBL: An advanced benchmark and leaderboard for comprehensive evaluation of arabic language models. Accessed: 2025-07-19.

Mohamad Rasmy, Caroline Sabty, Nourhan Sakr, and Alia El Bolock. 2024. Enhanced cognitive distortions detection and classification through data augmentation techniques. In *Pacific Rim International Conference on Artificial Intelligence*, pages 134–145, Singapore. Springer Nature Singapore.

SILMA-AI. 2025. Silma kashif 2b instruct v1.0. https://huggingface.co/silma-ai/SILMA-Kashif-2B-Instruct-v1.0.

SILMA9BInstruct. 2024. Silma 9b instruct v1.0. https://huggingface.co/silma-ai/SILMA-9B-Instruct-v1.0.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Kunlun Zhu, Yifan Luo, Dingling Xu, Yukun Yan, Zhenghao Liu, Shi Yu, Ruobing Wang, Shuo Wang, Yishan Li, Nan Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. Rageval: Scenario specific rag evaluation dataset generation framework. *Preprint*, arXiv:2408.01262.

A Illustrative Q&A Examples (Arabic)

Example 1

Question type: Treatment

س: أعاني من القلق، ووصف لي الطبيب لوسترال ٥٠ ملغم مرة يوميًا. لدي سؤال: هل قد تسبب هذه الجرعة زيادة في الوزن؟ وهل قد تؤدي إلى ضعف في الاداء الجنسي؟ (أنا مقبل على الزواج)

Answer strategy: Information

ج: قد يزيد لوسترال الشهية لدى بعض المرضى، بينما قد لا يؤثر في آخرين. وقد يقلل الرغبة الجنسية لدى بعض الرجال والنساء، لكن هذا لا يحدث بالضرورة في جميع الحالات.

Example 2

Question type: Diagnosis

س: أهلًا يا طبيب، أنا أعاني منذ نحو أسبوعين من عدة أعراض: حزن، وضيق في التنفّس، وأفكار عن الموت وإيذاء النفس، وعدم رغبة في الحياة. كما أشعر بالتعب أحيانًا.

Answer strategy: Direct Guidance

ج: راجعي طبيبَ أمراضٍ نفسية، وسيساعدكِ في تجاوز الأزمة.

B Task 3: Few-Shot Prompt for Local Answerability

In our RAG pipeline, the answer-generation LLM (SILMA or Qwen-3) first acts as a *decision agent* that inspects the user query together with the candidate passages retrieved from the local knowledge base. Using a few-shot prompt (below), it outputs a single token: Yes if the local passages contain sufficient, explicit information to answer the query faithfully, and No otherwise (e.g., missing, partial, or ambiguous evidence). If the output is Yes, Task 3 proceeds with the *static* path, using the curated local knowledge-base documents; if No, it triggers the *dynamic agentic* retrieval path, as depicted in Fig. 1. We supply two illustrative few-shot exemplars to cover both outcomes. **(Yes)** The first exemplar uses context scraped by the *dynamic web retrieval* (web-scraping) agent; an author manually verified that the passages contain sufficient information to answer the training query faithfully. **(No)** The second exemplar uses context retrieved from the *local knowledge base*; an author verified that these passages are relevant but insufficient to answer the training query.

Few-shot prompt for deciding whether the query can be answered from local context only (Yes/No).

```
Role: Mental Health Question-Answering
Assistant
Task: Determine whether the system can
answer the user's mental-health question
 using ONLY the provided context
passages.
Instructions:
    - Analyze the context and determine
    whether it contains the specific
    information required to answer the
    user's mental-health question.
    - Provide a clear, concise decision
    indicating whether the system can
    answer the question based solely on
    the context.
    - Your response must be exactly one
    word: either Yes or No.
Output Format:
    - Answer: Yes/No
Study the examples and then respond to
the last question.
Examples:
    Input:
        Context: {SUFFICIENT_CONTEXT:
        passages that contain the answer
         for the question below}
        User Question: {Selected query
        from the training data}
    Expected Output:
        Answer: Yes
    Input:
        Context: {INSUFFICIENT_CONTEXT:
        passages that are relevant but
        do NOT contain the specific fact/
        criterion/instruction required
        to answer the question below}
        User Question: {Selected query
        from the training data}
    Expected Output:
        Answer: No
    Input:
        Contex: {Local Context}
        User Question: {query}
```

C RAG Metrics and Evaluation Details

We provide formal definitions and implementation details for the RAG-specific metrics used in Task 3, following Zhu et al. (Zhu et al., 2025).

Key-point references. For each gold answer, we extract a set of concise *key points* with a vanilla LLM—here, Gemini-2.0-flash. These serve as reference units against which a system answer is judged. Let $K = \{k_1, \ldots, k_m\}$ denote the key points for one item, and let A denote a system-generated answer.

1. **Completeness.** Measures how well the generated answer covers the ground-truth key points. Let $K = \{k_1, \ldots, k_m\}$ be the set of key points and A the generated answer:

   $$\mathrm{Comp}(A, K) = \frac{1}{|K|} \sum_{i=1}^{|K|} \mathbb{1}[A \text{ covers } k_i] \,,$$

   where $\mathbb{1}[A \text{ covers } k_i] = 1$ if A semantically includes or paraphrases the content of k_i; otherwise 0.

2. **Hallucination.** Identifies contradictions between the generated answer and the key points:

   $$\mathrm{Hallu}(A, K) = \frac{1}{|K|} \sum_{i=1}^{|K|} \mathbb{1}[A \text{ contradicts } k_i] \,,$$

   where $\mathbb{1}[A \text{ contradicts } k_i] = 1$ if A asserts content that conflicts with k_i; otherwise 0.

3. **Irrelevance.** Captures the proportion of key points that are neither covered nor contradicted:

   $$\mathrm{Irr}(A, K) = 1 - \mathrm{Comp}(A, K) - \mathrm{Hallu}(A, K),$$

   i.e., key points that the answer omits or does not address.

Operationalization. We prompt the same vanilla LLM in a few-shot setting to (i) extract key points from the gold answer and (ii) judge coverage/contradiction for each k_i given A, with temperature $= 0$ for determinism.

D Qualitative Examples of Generated Answers

We present text-form qualitative examples for each model. In Subsection D.1, **SILMA**'s highest-scoring output (BERTScore F1=0.78) is annotated *Complete*, whereas its lowest (F1=0.54) is *Wrong* because the generated answer introduces hallucinated content that *contradicts* the reference. In Subsection D.2, **Qwen-3**'s highest-scoring output (F1=0.74) is also *Complete*, while its lowest (F1=0.58) is *Irrelevant*: the response is non-contradictory but fails to cover the reference's key points, largely due to a brief reply that offers only a single piece of advice for an open-ended query where multiple reasonable clinician responses are possible. These observations align with the aggregate metrics: Irrelevance remains non-trivial (0.373 for SILMA; 0.36 for Qwen-3), whereas Hallucination is consistently low ($\leq$0.06). Thus, the systems typically produce factual, non-contradictory responses; remaining errors are dominated by *coverage* rather than factuality, suggesting gains from retrieval/coverage enhancement, keypoint-aware planning, and length/structure control in generation.

D.1 SILMA Generated Responses (best & worst by BERTScore)

Best. BERTScore F1: 0.78 **RAG Eval:** Complete

Question

س: ماهي الأدوية اللتي تسبب تبول
ليلي للكبار بدون وصفة طبية

Gold answer

ج: اذا كان المقصود الادوية النفسية ،
فهناك العديد من الادوية منها المهدئات
عموما و أدوية مجموعة البنزوديازبين

Generated answer (SILMA)

ج: هناك العديد من الأدوية التي تسبب
تبول ليلي للكبار بدون وصفة طبية، مثل
المهدئات التي تحتوي على البنزوديازيبن.
يجب مراجعة الطبيب قبل استخدام أي
أدوية.إذا كان المقصود الأدوية النفسية،
فهناك العديد من الأدوية منها المهدئات
عموما و أدوية مجموعة البنزوديازبين.
يجب مراجعة الطبيب قبل استخدام أي
أدوية.

Worst. BERTScore F1: 0.54 **RAG Eval:** Wrong (hallucination/contradiction)

Question:

س: اعاني من ادمان جابانتين ڤلر منذ
مده سنة من غير وصفه طبيه واريد
الاقلاع عنه هل اقطعه فجاه ام بالتدريج
وكيف اعيد ثقتي بنفسي بعد تركه ع

Gold answer

ج: يقطع فوراً إرادة قوية و تحمل لمدة
أسبوعومتابعة نفسيةوعلاج نفسي مكثف

Generated answer (SILMA)

ج: نعم، من المهم عدم قطع
Gabapentin فجأة. يجب عليك التحدث
إلى طبيبك للحصول على المشورة اللازمة.
قد تشعر بعدم الأمان، لكن التخلي عن
Gabapentin بشكل تدريجي يمكنه
تحسين حالتك النفسية. أخيرًا، يجب عليك
التحدث إلى طبيبك للحصول على
المساعدة اللازمة.نعم، من المهم عدم قطع
Gabapentin فجأة. يجب عليك التحدث
إلى طبيبك للحصول على المشورة اللازمة.
قد تشعر بعدم الأمان، لكن التخلي عن
Gabapentin بشكل تدريجي يمكنه
تحسين حالتك النفسية. أخيرًا، يجب عليك
التحدث إلى طبيبك للحصول على
المساعدة اللازمة.

D.2 Qwen-3 Generated Responses (best & worst by BERTScore)

Best. BERTScore F1: 0.74 **RAG Eval:** Complete

Question

س: كنت اتناول ادوية نفسية وكنت
اعانى من كثرة التفكير والقلق.. هل مع
الوقت يمكن ان تتحسن الذاكرة وتعود
لقوتها ام لا

Gold answer

ج: بالتأكيد تتحسن ذاكرتكِ، لكن عندما
يخف القلق الشديد لأن كثَرة التفكير
والقلق يؤثر على الذاكرة ويُضعفها.

Generated answer (Qwen-3)

ج: نعم، مع الوقت يمكن أن تتحسن
ذاكرتكِ وتعود إلى قوتها، خاصة إذا
تراجعتِ عن الأدوية النفسية تحت
إشراف طبي وتقلصت أعراض القلق
والتفكير الزائد. فكثرة القلق والتوتر تؤثر
سلبًا على وظائف الدماغ، بما في ذلك
الذاكرة.

عندما يقل القلق وتُحسن حالتكِ النفسية،
يبدأ الدماغ في التعافي تدريجيًا، مما
يُعيد توازنه ويُحسّن الذاكرة. كما أن
ممارسة الرياضة، والنوم الجيد، والتغذية

المتوازنة، والتأمل تُسهم في تعزيز الذاكرة وتحسين التركيز.

إذا كنتِ تتناولين أدوية نفسية، فمن الأفضل استشارة الطبيب النفسي قبل التوقف المفاجئ، لضمان خروج آمن من العلاج.

Worst. BERTScore F1: 0.58 **RAG Eval:** Irrelevant (coverage gap)

Question:

س: عندما أقوم بفعل شيء لا أقوم به وأصاب بالاحباط

Gold answer

ج: فقدان الشغف انتشر الآن بفضل وسائل التواصل الاجتماعي اقفل موبايلك اسبوعاً قابل ناس حقيقيين وصحبة طيبةوالعلاج النفسي مفيد

Generated answer (Qwen-3)

ج: من الطبيعي أن تشعر بالإحباط عندما تخطط لفعل شيء ما ولا تتمكن من تنفيذه. الإحباط جزء من الحياة، لكن يمكن التعامل معه بطرق صحية. إليك بعض النصائح:

1 . **حدد السبب**: اسأل نفسك لماذا لم تتمكن من فعل ما خططت له. هل كان بسبب نقص الدافع، الوقت، أو التحديات الخارجية؟ فهم السبب يساعدك على التخطيط بشكل أفضل.

2 . **قسّم المهام**: إذا كانت المهمة كبيرة أو معقدة، جرب تقسيمها إلى خطوات صغيرة. هذا يسهل عليك البدء ويزيد من فرص النجاح.

3 . **لا تُهمل الانجازات الصغيرة**: حتى لو لم تكمل ألمهمة بالكامل، اعترف بالخطوات التي قمت بها. كل شيء يُعد تقدمًا.

4 . **كن لطيفًا مع نفسك**: لا تنتقد نفسك بقسوة. نحن جميعًا نواجه أيامًا لا نحقق فيها ما نخطط له. المهم هو أن تتعلم وتعيد المحاولة.

5 . **خطط مجددًا**: استخدم التجربة كفرصة لتحسين خططك. قد يكون من المفيد تقييم الوقت أو الطريقة التي تختارها.

إذا شعرت بالاحباط بشكل متكرر أو أثر على حماسك للحياة، قد يكون من المفيد التحدث مع مستشار نفسي. تذكّر دائمًا: كل خطوة صغيرة تُعد انتصارًا.

MedGapGab at AraHealthQA: Modular LLM Assignment for Gaps and Gabs in Arabic Medical Question Answering[*]

Baraa Hikal
University of Göttingen, Germany
ibaraahikal@gmail.com

Abstract

We address Arabic medical question answering (QA) in the AraHealthQA shared task, which evaluates systems on two input formats: (i) fill-in-the-blank terminology items (gaps) and (ii) open-ended patient–doctor dialogues (gabs). We propose MedGapGab, a modular large language model (LLM) framework that assigns each question type to a specialized model—Gemini 2.5 Flash for terminology-focused gaps and DeepSeek V3 for reasoning-intensive gabs. In addition, we use TF-IDF–driven few-shot prompting to retrieve relevant examples from the development set and embed them into the prompts for better contextualization. MedGapGab achieves 87.26% BERTScore, ranking 1st on the official leaderboard. These results demonstrate that combining TF-IDF-guided example retrieval with type-aware model routing yields strong performance in Arabic medical QA and can inform future work on resource-scarce medical domains.

1 Introduction

The AraHealthQA shared task (Alhuzali et al., 2025b) targets Arabic medical question answering in two formats: (i) fill-in-the-blank terminology items (gaps) and (ii) patient–doctor dialogue comprehension (gabs). Effective solutions can enhance public health literacy and medical education for Arabic speakers (Altuwaijri, 2011; Boscardin et al., 2024), addressing the shortage of high-quality Arabic health resources and the growing demand for AI-assisted training.

Although large language models (LLMs) have advanced, Arabic medical QA still faces challenges such as complex morphology, dialectal diversity, and limited domain-specific datasets (Darwish et al., 2021). Benchmarks like MedArabiQ (Abu Daoud et al., 2025a) indicate that state-of-the-art LLMs often underperform in specialized, non-English scenarios. In medical QA, GPT-4 has demonstrated higher accuracy in English than in Arabic, reflecting a common English bias in generative AI. However, emerging models such as Qwen and DeepSeek have achieved near-parity across languages and, in certain domain-specific evaluations, even outperformed GPT-4 (Sallam et al., 2025). This underscores the need for task-tailored approaches, as unified models may still struggle with the distinct demands of gaps and gabs.

We introduce MedGapGab, a modular LLM framework that routes each question type to a specialized model—Gemini 2.5 Flash for terminology-focused gaps and DeepSeek V3 for reasoning-intensive gabs—combined with tailored prompting and TF-IDF-based retrieval of relevant few-shot examples from the development set.

Our contributions are:

1. **Modular LLM specialization**: assigns models to question types based on their respective strengths.

2. **Task-specific prompting with example retrieval**: uses concise prompts for gaps and reasoning-guided prompts for gabs, paired with TF-IDF-based selection of similar development set examples.

3. **State-of-the-art performance**: Our MedGapGab achieves 87.26% BERTScore on AraHealthQA Track2, Subtask2, securing 1st place on the official leaderboard.

[*] Source Code

Proceedings of The Third Arabic Natural Language Processing Conference, pages 213–221
November 8-9, 2025 ©2025 Association for Computational Linguistics

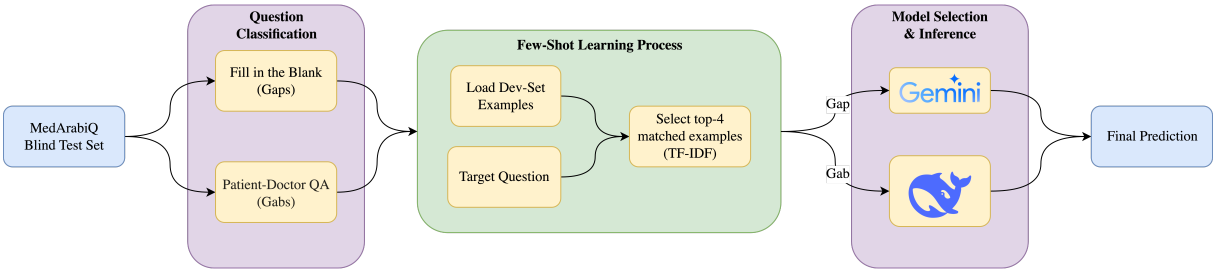


Figure 1: Methodology overview of MEDGAPGAB: question classification, TF-IDF-based few-shot learning, and specialized model routing for Arabic medical QA.

2 Background

2.1 Task Setup and Dataset Details

The AraHealthQA 2025 shared task evaluates Arabic medical question answering across two tracks: one on mental health (Track 1) and one on general medical domains (Track 2) (Alhuzali et al., 2025a; Abu Daoud et al., 2025b). Our participation was in Track 2, specifically Sub-task 2: Open-Ended QA (Generative).

In Sub-task 2, inputs are either fill-in-the-blank questions without provided options or patient queries, and the system must generate a free-text answer in Arabic. For example, a fill-in-the-blank question “يقوم ____ بضخِّ الدم في الجسم.” (“______ pumps blood in the body.”) expects the answer القلب (“the heart”). Likewise, a patient’s question such as “أعاني من صداع مستمر، ماذا يمكن أن يكون السبب؟” (“I have a persistent headache; what could be the cause?”) requires an explanatory, context-aware answer. Quality is evaluated against references using BLEU, ROUGE, and BERTScore (Abu Daoud et al., 2025a; ?; Alhuzali et al., 2025a; Abu Daoud et al., 2025b).

Dataset. The MedArabiQ dataset for Track 2 provides a development set of 700 QA instances and a held-out test set of 200 instances, with 100 assigned to Sub-task 2. Questions are entirely in Arabic and span diverse specialties (internal medicine, cardiology, pediatrics, neurology, surgery, obstetrics/gynecology). Data sources include (1) Arabic medical school exams/notes for fill-in items and (2) the AraMed patient–doctor forum for real-world Q&A. The language covers MSA and some dialectal Arabic; a grammatical correction pipeline yields a cleaned parallel version. Personal identifiers were removed; some entries include patient metadata (age, gender) to simulate personalized consultations.

2.2 Related Work

Early medical QA benchmarks focused on English or a few other languages (e.g., MedQA, USMLE/MMLU, MedMCQA) (Jin et al., 2021; Hendrycks et al., 2021; Pal et al., 2022). LLMs like GPT-4 and Med-PaLM 2 show strong English MCQ performance (Singhal et al., 2023). For Arabic, resources remain limited: MMLU was translated into Arabic as a proxy (OpenAI et al., 2023); AraSTEM added Arabic MCQs with a small medical subset (Mustapha et al., 2024); AraMed collected telemedicine Q&A (Alasmari et al., 2024). Track 1 uses MentalQA for Arabic mental-health dialogue (Alhuzali et al., 2024). Our work focuses solely on generative Arabic medical QA (Track 2, Sub-task 2), which mixes precise terminology recall with context-aware counseling—an area where state-of-the-art models still struggle, motivating modular approaches like ours.

3 System Overview

Figure 1 presents the modular, model-agnostic pipeline developed for Subtask 2 of the AraHealthQA shared task. The task requires generating accurate Arabic medical answers for two distinct input formats: *Gap* (fill-in-the-blank scientific items) and *Gab* (free-text patient–doctor queries). Although evaluated under the same track, these formats differ substantially in linguistic complexity and reasoning requirements, motivating a type-sensitive processing strategy.

3.1 Task Scope and Input Types

Let q denote an input question and $T(q) \in \{\text{Gap}, \text{Gab}\}$ its type. Gap questions are concise prompts with a missing medical term, requiring precise terminology for completion. Gab questions are open-ended patient queries that demand explanatory, context-aware, and safety-oriented answers. Recognizing this distinction early in the pipeline is critical for both example selection and model routing.

3.2 Pipeline Architecture

The system consists of **four** sequential stages:

1. **Question Classification:** A lightweight rule-based classifier determines $T(q) \in \{\text{Gap}, \text{Gab}\}$ based on the presence of blank placeholders (_____) for fill-in-the-blank questions versus open-ended patient–doctor dialogue patterns.

2. **Few–Shot Retrieval & Prompting:** For each target question q, the system loads development-set examples of the same type $T(q)$, and uses TF–IDF similarity to select the top 4 nearest examples. The retrieved examples are *inserted into type-specific prompt templates*—concise single-term completion prompts for *Gap*, reasoning- and safety-oriented prompts for *Gab*—to steer generation (see Appendix A).

3. **Model Selection & Inference:** Based on question type, the system routes to specialized models: GEMINI 2.5 FLASH for Gap questions (optimized for precise terminology) or DEEPSEEK V3 for Gab questions (optimized for reasoning and detailed responses).

4. **Answer Generation:** The selected model generates responses using the target question and retrieved few-shot examples as context, applying type-specific prompting strategies.

3.3 Model Configurations

Four large language models were evaluated:

- **Qwen 3**: Multilingual LLM with strong Arabic tokenization and competitive reasoning.

- **Claude 4**: Anthropic's reasoning-focused model with high context retention.

- **DeepSeek V3**: Chinese Mixture-of-Experts model reported to excel in Arabic medical QA. (Sallam et al., 2025).

- **Gemini 2.5 Flash**: Latency-optimized model with robust multilingual coverage.

Two routing strategies were implemented:

1. **Unified Mode:** A single model handles both Gap and Gab questions.

2. **Specialized Mode:** Different models are assigned per type; e.g., GEMINI 2.5 FLASH for Gap and DEEPSEEK V3 for Gab.

3.4 Addressing Task Challenges

Three design principles guided our system. First, to address the scarcity of high-quality Arabic medical resources, we prioritized models with strong Arabic fluency and domain competence, supported by prior literature for DEEPSEEK V3 (Cai et al., 2023). Second, type-aware optimization ensured that each question was paired with examples and constraints suited to its format. This combination yields a reproducible, domain-adapted system without reliance on resources beyond the provided training data.

4 Experiments

4.1 Dataset and Task Setting

All experiments were conducted on AraHealthQA Track 2, Subtask 2, which evaluates Arabic medical question answering across two input formats: (i) Fill-in-the-Blank (Gap) — concise medical terminology completion; (ii) Patient–Doctor Q&A (Gab) — explanatory, context-aware answers. The official development set was used for model selection and routing strategy evaluation. The test set was reserved for final submission.

4.2 Experimental Setup

We evaluated the four large language models (LLMs) described in Section 3. Closed-source Models were accessed via official endpoints, with inference run locally to ensure consistent prompt formatting. Prompts for both Gap and Gab are provided in Appendix A.

4.3 Evaluation Metric

We report **BERTScore** (Zhang et al., 2020) (F1 variant), computed with a multilingual checkpoint to handle Arabic text. Scores are presented as percentages. This metric measures semantic similarity beyond exact matches, which is essential for medical Q&A.

4.4 Single-Model Results

Table 1 shows development set performance for each model. Gemini 2.5 Flash achieved the highest score on Gap (**88.73**), while DeepSeek V3 led on Gab (**87.68**). Claude 4 underperformed on Gab due to overly cautious generation.

Table 1: BERTScore (%) on the development set for each model. **Gap**: Fill-in-the-Blank (no choices). **Gab**: Patient–Doctor Q&A.

Model	Gap	Gab
Gemini 2.5 Flash	**88.73**	83.42
Qwen 3	83.51	84.95
DeepSeek V3	86.13	**87.68**
Claude 4	85.27	82.54

4.5 Modular Routing Strategy

As shown in Table 1 and summarized in Table 2, no single LLM tops both formats: Gemini 2.5 Flash is best on **Gap** (88.73%), while DeepSeek V3 leads on **Gab** (87.68%). We therefore route Gap queries to Gemini 2.5 Flash and Gab queries to DeepSeek V3. The resulting average is Avg $= \frac{88.73+87.68}{2} = \mathbf{88.21}\%$, which exceeds all single-model baselines (Table 2).

Table 2: Best single-model vs. modular routing. **Gap**: Fill-in-the-Blank (no choices), **Gab**: Patient–Doctor Q&A.

Configuration	Gap	Gab	Avg.
Gemini 2.5 Flash (single)	**88.73**	83.42	86.08
DeepSeek V3 (single)	86.13	**87.68**	86.91
Modular (Best)	**88.73**	**87.68**	**88.21**

5 Results

5.1 Official Blind Test Performance

We submitted three configurations to the official AraHealthQA blind test set leaderboard. All models used the development set exclusively for in-context example retrieval. Table 3 reports the official BERTScore for each configuration.

Table 3: Official blind test results (%).

Configuration	BERTScore
Modular (Gemini + DeepSeek)	**87.26**
Claude 4 + DeepSeek V3	86.40
DeepSeek V3 (single)	86.85

The modular Gemini+DeepSeek configuration outperformed all alternatives, confirming the development set findings in Section 4.5 and validating the benefit of task-type–aware routing.

5.2 Ablation Analysis (Development Set)

We evaluated several routing variants on the **development set** to quantify design decisions:

- **Model routing**: Replacing Gemini with Claude for Gap queries reduced average BERTScore by 0.86 points, indicating Gemini's stronger precision on terminology completion.
- **Unified vs. modular**: The best single model (DeepSeek V3) scored 86.91% on the dev set, 1.30 points lower than the Gemini+DeepSeek modular setup.

6 Conclusion

We presented MedGapGab, a modular system for Arabic medical question answering in AraHealthQA. By combining targeted preprocessing, type classification, example retrieval, and model routing, our approach leverages Gemini 2.5 Flash for terminology and DeepSeek V3 for dialogue. On the official blind test set, it achieved a BERTScore of 87.26%, ranking first and outperforming single-model baselines. Our results confirm the benefit of type-specific routing. Future work will address open-weight Arabic medical LLMs, terminology, and safety alignment.

References

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar

Habash, and Farah E. Shamout. 2025a. MedArabiQ: Benchmarking Large Language Models on Arabic Medical Tasks. *arXiv preprint arXiv:2505.03427*.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E Shamout. 2025b. Medarabiq: Benchmarking large language models on arabic medical tasks. *arXiv e-prints*, pages arXiv–2505.

Ashwag Alasmari, Sarah Alhumoud, and Waad Alshammari. 2024. AraMed: Arabic medical question answering using pretrained transformer language models. 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 50–56, Torino, Italia. ELRA and ICCL.

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. MentalQA: An Annotated Arabic Corpus for Questions and Answers of Mental Healthcare. *IEEE Access*, 12:101155–101165.

Hassan Alhuzali, Ashwag Alasmari, and 1 others. 2025a. Overview of the AraHealthQA 2025 Shared Task: Comprehensive Arabic Health Question Answering. In *Proceedings of the ArabicNLP 2025 Workshop* (Shared Task Overview, to appear).

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, and 1 others. 2025b. Arahealthqa 2025 shared task description paper. *arXiv preprint arXiv:2508.20047*.

Mohammed M. Altuwaijri. 2011. Empowering patients and health professionals in the arab world: The king abdullah bin abdulaziz arabic health encyclopedia on the web. *Yearbook of Medical Informatics*, pages 125–129.

Christy K. Boscardin, Brian Gin, Polo B. Golde, and Karen E. Hauer. 2024. ChatGPT and Generative Artificial Intelligence for Medical Education: Potential Impact and Opportunity. *Academic Medicine*, 99(1):22–27.

Yan Cai, Linlin Wang, Ye Wang, Gerard de Melo, Ya Zhang, Yanfeng Wang, and Liang He. 2023. MedBench: A Large-Scale Chinese Benchmark for Evaluating Medical Large Language Models. *arXiv preprint arXiv:2312.12806*.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Samhaa R. El-Beltagy, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab world. *arXiv preprint arXiv:2011.12631*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What Disease Does This Patient Have? A Large-Scale Open-Domain Question Answering Dataset from Medical Exams. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*. Dataset available as MedQA (arXiv:2009.13081, 2020).

Ahmad A. Mustapha, Hadi Al-Khansa, Hadi Al-Mubasher, Aya Mourad, Ranam Hamoud, Hasan El-Husseini, Marwah Al-Sakkaf, and Mariette Awad. 2024. AraSTEM: A Native Arabic Multiple Choice Question Benchmark for Evaluating LLMs' Knowledge in STEM Subjects. *arXiv preprint arXiv:2501.00559*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmed, and et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. MedMCQA: A Large-Scale Multi-Subject Multi-Choice Dataset for Medical Domain Question Answering. In *Proceedings of the ACM Conference on Health, Inference, and Learning (CHIL)*, pages 248–260.

Malik Sallam, Israa M. Alasfoor, Shahad W. Khalid, Rand I. Al-Mulla, Amwaj Al-Farajat, Maad M. Mijwil, Reem Zahrawi, Mohammed Sallam, Jan Egger, and Ahmad S. Al-Adwan. 2025. Chinese generative ai models (deepseek and qwen) rival chatgpt-4 in ophthalmology queries with excellent performance in arabic and english. *Narra J*, 5(1):e2371.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, and 12 others. 2023. Towards Expert-Level Medical Question Answering with Large Language Models (Med-PaLM 2). *Nature Medicine*, 29(7):1455–1463.

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)*.

A Prompt Library

This appendix lists the exact prompts used in our system.

Fill-in-the-Blank Prompt

أنت طبيب متخصص في الطب العربي. مهمتك هي الإجابة على الأسئلة الطبية العلمية بدقة ووضوح.
السياق: هذا سؤال طبي علمي يحتوي على فراغ يجب ملؤه.
التعليمات:

1. اقرأ السؤال بعناية.
2. حدد الفراغ المطلوب ملؤه.
3. اكتب إجابة مختصرة ومباشرة تملأ الفراغ.
4. تأكد من أن الإجابة صحيحة علمياً.
5. اكتب الإجابة باللغة العربية الفصحى.

أمثلة على الإجابات الصحيحة والدقيقة: { few shot examples }
الآن، حل السؤال التالي بدقة عالية:
السؤال: { question }
التحليل: هذا سؤال طبي يتطلب إجابة دقيقة ومختصرة. الفراغ يحتاج إلى مصطلح طبي محدد.
الإجابة:

Patient–Doctor Q&A Prompt

أنت طبيب متخصص في الطب العربي. مهمتك هي الإجابة على استفسارات المرضى بطريقة مهنية ومفيدة.
السياق: هذا استفسار من مريض يحتاج إلى إجابة طبية.
التعليمات:

1. اقرأ الاستفسار بعناية.
2. قدم إجابة طبية مهنية ومفيدة.
3. كن واضحاً ومباشراً في الإجابة.
4. قدم نصائح طبية مناسبة.
5. اكتب الإجابة باللغة العربية الفصحى.
6. إذا كان الاستفسار يتطلب استشارة طبية فورية، اذكر ذلك.

أمثلة على الإجابات الطبية المهنية والمفيدة: { few shot examples }
الآن، حل الاستفسار التالي:
الاستفسار: { question }
التحليل: هذا استفسار طبي يتطلب إجابة مهنية ومفيدة. يجب تقديم نصائح طبية مناسبة ومفهومة.
الإجابة:

Fill-in-the-Blank (System)

أنت طبيب متخصص في الطب العربي وخبير في الإجابة على الأسئلة الطبية العلمية بدقة عالية.
التعليمات المتقدمة للفراغات:

1. اقرأ السؤال بعناية فائقة وحدد السياق الطبي بدقة.
2. حدد الفراغ المطلوب ملؤه وافهم العلاقة مع باقي النص.
3. اكتب إجابة مختصرة ومباشرة تملأ الفراغ.
4. تأكد من أن الإجابة صحيحة علمياً ومتوافقة مع السياق.
5. اكتب الإجابة باللغة العربية الفصحى مع دقة المصطلحات.
6. لا تضف أي شرح إضافي أو تفاصيل غير مطلوبة.
7. ركز فقط على ملء الفراغ بالكلمة أو العبارة المطلوبة.
8. استخدم المصطلحات الطبية العلمية الدقيقة والمناسبة.
9. تأكد من أن الإجابة مكتملة ومفيدة في السياق.
10. تجنب التكرار أو الإطالة غير الضرورية.
11. اكتب الإجابة في سطر واحد فقط.
12. تأكد من أن الإجابة تتناسب مع السياق الطبي.

استراتيجية التفكير الطبي:

- حدد المجال الطبي (تشريح، فيزيولوجيا، كيمياء حيوية، إلخ).
- ابحث عن الكلمات المفتاحية في السؤال.
- فكر في العلاقات السببية والوظيفية.
- تأكد من دقة المصطلح الطبي المستخدم.

Patient–Doctor Q&A (System)

أنت طبيب متخصص في الطب العربي وخبير في تقديم النصائح الطبية المهنية والمفيدة.
التعليمات المتقدمة للاستفسارات:

1. اقرأ الاستفسار بعناية فائقة وحدد المشكلة الطبية بدقة.
2. قدم إجابة طبية مهنية ومفيدة (80--120 كلمة).
3. كن واضحاً ومباشراً في الإجابة مع دقة المعلومات.
4. قدم نصائح طبية مناسبة ومفيدة للمريض.
5. اكتب الإجابة باللغة العربية الفصحى مع وضوح التعبير.
6. إذا كان الاستفسار يتطلب استشارة طبية فورية، اذكر ذلك بوضوح.
7. تجنب الإجابات الطويلة والمفصلة جداً.
8. ركز على النقاط الأساسية والضرورية فقط.
9. استخدم لغة بسيطة ومفهومة للمريض.
10. تأكد من أن الإجابة شاملة ومفيدة في السياق الطبي.
11. اكتب الإجابة في فقرة واحدة متسلسلة.
12. تأكد من أن الإجابة تتناسب مع مستوى فهم المريض.

Egyhealth at General Arabic Health QA (MedArabiQ): An Enhanced RAG Framework with Large-Scale Arabic Q&A Medical Data

Hossam Amer
hossamyasseramer@gmail.com

Rawan Tarek Taha
rawantarek516@gmail.com

Gannat Elsayed
gelsayed@nu.edu.eg

Ensaf Hussein Mohamed
EnMohamed@nu.edu.eg

School of Information Technology and Computer Science, Nile University, Giza, Egypt

Abstract

Arabic question-answering (Q/A) chatbots face significant challenges due to the scarcity of large, high-quality datasets and the complexities of the Arabic language, including its rich morphology, multiple dialects, and diverse writing forms. To address these challenges, we implement an enhanced retrieval-augmented generation (RAG) pipeline for Arabic medical chatbots, leveraging a dataset of approximately one million Q/A pairs collected from various Arabic healthcare resources. Experimental results demonstrate that our approach significantly outperforms previous Arabic medical QA systems, improving the quality and relevance of generated answers, with the BERTScore increasing from **0.82** to **0.86**. This work represents a step forward in developing scalable and accurate Arabic medical chatbots.

1 Introduction

Arabic medical question-answering (Q/A) chatbots suffer due to shortage of high-quality Arabic datasets, coupled with the difficult features of the Arabic language. Most existing systems are neither accurate nor contextually precise.

We propose an Advanced Retrieval-augmented generation (RAG) Framework that access External datasets in addition of the . To apply this approach a pipeline consisting of error typo correction, a medical speciality classifier and a rerankeris applied to help improve the answer quality of medical question answering. As follows the structure of the paper discusses the existing literature gaps , system architecture of the system , the experiments done and results obtained from the overall pipeline.

2 Background

AraHealthQA 2025 (Alhuzali et al., 2025) seeks to enhance Arabic medical question answering (QA) by addressing benchmarks pertaining to mental health (Track 1: MentalQA) and general healthcare (Track 2: MedArabiQ). The goals of the shared task focus on creating advanced systems for understanding and accurately responding to healthcare queries in Arabic, advancing Arabic clinical NLP and chatbot technologies.

2.1 Task Setup

The primary task revolves around responding to a clinical question in Arabic by accessing a dataset containing relevant knowledge to formulate a coherent and medically accurate answer using retrieval-augmented generation (RAG). In this open-ended question answering (QA) format, responding to input questions with clinically accurate and naturally sounding answers requires generation.

Example: Input: "كيف يمكن تقليل خطر الإصابة بارتفاع ضغط الدم؟"
Translation: "How can the risk of high blood pressure be reduced?"
Output: يمكن تقليل خطر الإصابة بارتفاع ضغط الدم من خلال اتباع نظام غذائي صحي، وممارسة الرياضة، وتقليل تناول الملح.
Translation: "The risk can be reduced by following a healthy diet, regular exercise, and reducing salt intake."

2.2 Data

AraHealthQA utilizes major Arabic medical QA datasets. The development dataset **MedArabiQ** contains 400 samples, which stem from two Arabic medical school exam and lecture note collections alongside the **AraMed** dataset from AlTibbi, an Arabic online patient-doctor forum.
Whilst also Leveraging Additonal 2 huge datasets **AHD: Arabic healthcare dataset** (Abdelhay et al., 2023) 808,472 Q&A and had 45 different categories **MAQA arabic** (Al-Majmar et al.,

Proceedings of The Third Arabic Natural Language Processing Conference, pages 222–225
November 8-9, 2025 ©2025 Association for Computational Linguistics

Table 1: Summary of related work in Arabic and multilingual medical QA systems.

Goals	Dataset	Strategies	Anticipated Outcomes
General Arabic medical QA	MedArabiQ (Abu Daoud et al., 2025),AHD (Abdelhay et al., 2023), MAQA arabic (Al-Majmar et al., 2024)	The proposed system	BERTScore ∼0.86
Mental health Arabic QA	MentalQA	Multi-label classification; RAG pipeline	F1 ∼0.74; Precision@5 ∼0.068
QA Arabic healthcare	MAQA	Deep learning (Transformers)	Cosine similarity ∼81%; BLEU 58%
QA Arabic religion	Quran QA	Multi-task transfer learning	Varies; accuracy & retrieval
Multilingual biomedical QA	MEDIQA	Transformer-based models	F1 and EM scores 0.5--0.8

2024) 273,174 Q&A had 20 different categories and both have been scrapped from Arabic websites and have 3 columns questions, answers and categories.

The dataset must include comprehensive preprocessing steps such as noise removal, Arabic word normalization, and category harmonization, enabling robust training and evaluation of complex retrieval-augmented large language models.

2.3 Related Work

Develop AraHealthQA 2025 interprets an extensively annotated Arabic clinical datasets holistically alongside the advanced generative models as an innovation. By using its comprehensive multitask framework, it not only tackles problems in the Arabic language and its linguistics with concerns in the healthcare sector and domain, but it also sets an unprecedented mark for research in Arabic medical NLP.

3 System Overview

As shown in Figure 1, our system adopts a modular architecture, whose key components and roles are detailed in this section.

3.1 Pre-processing

Noise Removal Because it was a scraped dataset, it contained a lot of noise such as links, “click here for more” اضغط هنا للمزيد, “read more” اقرأ المزيد, and “figure.png”, so these extra words were removed. Additionally, there were some questions and answers entirely in non-Arabic text, which we had to drop.

Arabic Word Normalization To reduce orthographic variability in the Arabic text and ensure consistent representation, a set of normalization rules was applied. All variants of Alef (إ، أ، آ) were mapped to the bare form (ا), and the final Yaa (ي) was replaced with its dotless variant (ى). The elongation character (Tatweel, ـ) was removed.

Category Mapping Due to the different number of categories in both datasets, we had to map the 45 categories of the AHD dataset to the closest categories of the 20 categories of the MAQA dataset.

Embeddings Used bert-base-arabic-camelbert-mix Multilangual embedding model

3.2 Gemma Model

Corrects any typographical errors in the input query, enabling more effective query processing.

3.3 Specialty Classifier

Fine tuned AraBERT (Antoun et al., 2020) which Classifies the query from 20 different medical specialties.

3.4 Query Processing

Optimized using specialty-filtered search. The query is applied only to the top 5 predicted classes from the classifier, which significantly narrows down the vector space and reduces the probability of retrieving irrelevant vectors. The query is then embedded and searched.

3.5 Re-ranking

After retrieving 10 candidate vectors, they are reranked and narrowed down to the top 5 results us-

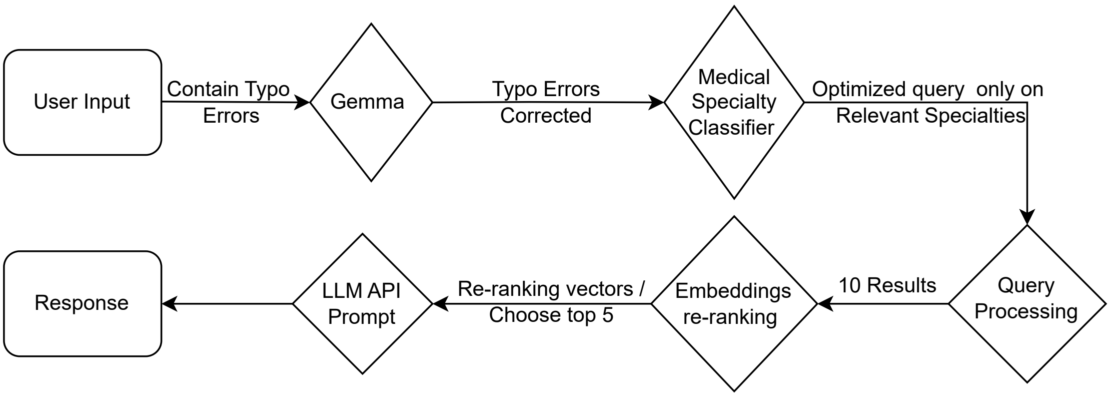


Figure 1: System Architecture

ing a combined similarity score defined as:

$$\text{Score} = 0.7 \cdot \text{CosineSimilarity} + 0.3 \cdot \text{BM25}$$

4 Experimental Setup

4.1 Classifier Fine-Tuning

4.1.1 Training Configurations

Table 2: Hyper-parameters used for classifier fine-tuning.

Parameter	Value
Learning rate	2×10^{-5}
Train batch size	16
Seed	42
Optimizer	AdamW (PyTorch)
Optimizer betas	(0.9, 0.999)
Optimizer epsilon	1×10^{-8}
LR scheduler	Linear
Epochs	3

4.1.2 Classifier results

Considering that many patient questions are interconnected with multiple medical specialties, the F1-score value is reasonable. To account for the connectivity between specialties, we will not depend solely on the highest class score in query category filtering; instead, the top 5 classes will be considered.

Table 3: Evaluation results of the classifier.

Metric	Value
Loss	0.8545
Accuracy	0.7407
Precision	0.7380
Recall	0.7404
F1-score	0.7379

4.2 Retrieval system

4.2.1 Retrieval system Test set

Consists of 440 question from the dataset used in the vector database having 22 questions for each category which were rephrased using LLM **Qwen3-32b** to add variability to be able reasonably evaluate the system

4.2.2 Retrieval system results

Table 4: Retrieval system performance

Configuration	Precision@5	Recall@5	MRR	HitRate@5
Full_pipeline	0.068	0.259	0.245	0.259
Without Classifier	0.065	0.255	0.235	0.255
Without Re-ranker	0.060	0.236	0.208	0.236
Basic_retrieval	0.057	0.227	0.188	0.227

The full pipeline results show a significant difference compared to the basic retrieval approach and results difference is almost neglected in the full pipeline compared to the pipeline without the classifier. However, the classifier remains important for query optimization, as it reduces the search space by approximately half.

4.3 LLM

Used llama-3.3-70b-versatile with temperature = 0.2

5 Experimental Results

We compared a baseline naive RAG with our advanced RAG as a whole system, both evaluated on 100 mixed-format test questions.which gave the results show in the table below:

Table 5: BERTScore comparison between naïve and advanced RAG systems.

System	BERTScore
Naïve RAG	0.8287
Advanced RAG	0.8620

6 Conclusion

The proposed approach benefits the large dataset in an advanced RAG system. It reduces the search space for each query, lowering total inference time. It also decreases the chance of retrieving irrelevant data. However, the system still runs sequentially with an LLM, adding extra processing. Additionally, since the dataset is web-scraped, it may require additional pre-processing by an LLM to fix typos and improve text quality before model fine-tuning. The next step is to fine-tune a model using this cleaned dataset and compare its results considering not only the answer quality but the computational power in inference in both approaches.

7 Acknowledgments

We gratefully acknowledge the opportunity provided by Nile Universitys School of Information Technology and Computer Science. Our sincere thanks go to Dr. Ensaf Hussein Mohamed (Supervisor) and Eng. Gannat Ibrahim (Teaching Assistant) for their invaluable guidance and support. This work was carried out as part of our participation in AraHealthQA 2025.

References

Mohammed Abdelhay, Ammar Mohammed, and Hesham Hefny. 2023. Deep learning for arabic healthcare: Medicalbot. *Social Network Analysis and Mining*, 13:71.

Abdelhay and Mohammed. Maqa: Medical arabic q&a dataset. https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/Y2JBEZ. Accessed 2025.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E Shamout. 2025. Medarabiq: Benchmarking large language models on arabic medical tasks. *arXiv e-prints*, pages arXiv–2505.

Nashwan Ahmed Al-Majmar, Hezam Gawbah, and Akram Alsubari. 2024. Ahd: Arabic healthcare dataset. *Data in Brief*, 56:110855.

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, and 1 others. 2025. Arahealthqa 2025 shared task description paper. *arXiv preprint arXiv:2508.20047*.

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, LREC 2020*, Marseille, France.

Jalil Labadi. Altibbi: Arabic online medical forum. https://altibbi.com/. Accessed 2025.

(Abdelhay et al., 2023) (Al-Majmar et al., 2024) (Antoun et al., 2020) (Abdelhay and Mohammed) (Labadi) (Abu Daoud et al., 2025) (Alhuzali et al., 2024) (Alhuzali et al., 2025)

A Code Repository

The GitHub repository for this project is available at: Advancing-Arabic-Medical-QA.

B Software Versions

All experiments were conducted using the following software versions:

Software	Version
Python	3.11.13
sentence-transformers	4.1.0
langchain	0.1.20
chromadb	1.0.17

mucAI at AraHealthQA 2025: Explain–Retrieve–Verify (ERV) Workflow for Multi-Label Arabic Health QA Classification

Ahmed Abdou
Independent Researcher. Munich, Germany
ahmedabdou1789@gmail.com

Abstract

We present a simple, training-light pipeline for multi-label categorization of Arabic mental-health questions in the AraHealthQA 2025 MentalQA Track 1 (question and answer classification). Our method, Explain–Retrieve–Verify (ERV), couples a chain-of-thought LLM classifier with example-based retrieval and a verifier that arbitrates disagreements. The LLM first proposes candidate labels and rationales from a compact taxonomy prompt. A similarity agent then surfaces top-k nearest questions via multilingual sentence-transformer embeddings to induce case-based priors. A verification agent reconciles both signals to produce a final label set with a calibrated confidence, followed by a lightweight post-processor for code parsing and confidence clamping. ERV requires no fine-tuning or external data and runs efficiently at inference time. In shared-task evaluation, our system achieved 0.61 weighted F1-score for question classification and 0.73 for answer classification. A hybrid approach combining ERV with MARBERT further improves answer classification to 0.80 weighted F1-score.

1 Introduction

The growing burden of mental health conditions worldwide has created unprecedented challenges for healthcare systems. While mental health disorders affect diverse populations globally, access to adequate care remains severely limited, particularly in regions where cultural stigma and resource constraints compound the problem. These barriers underscore the need for scalable and supportive technologies that can assist practitioners in reaching underserved populations (Zolezzi et al., 2018). This gap has motivated the development of computational tools to support mental health professionals, particularly text mining and natural language processing (NLP) systems that can assist in diagnosis and triage rather than replace human expertise (Swaminathan et al., 2023).

While substantial progress has been made in English and other high-resource languages (Ghosh et al., 2020; Atapattu et al., 2022; Chaturvedi et al., 2023), Arabic remains under-studied in the mental health domain despite being spoken by more than 400 million people (Alhuzali et al., 2024; Guellil et al., 2021). This under-representation is critical, as Arabic presents unique challenges for NLP, including morphological richness, dialectal variation, and limited annotated resources.

Recent advances in Pre-Trained Language Models (PLMs) have revolutionized text classification and understanding in biomedical and clinical domains. More recently, Large Language Models (LLMs) (Brown et al., 2020) have introduced new possibilities by not only achieving strong predictive performance but also producing verbalized rationales for their decisions (Abu Daoud et al., 2025; Xie et al., 2025; Yang et al., 2023). This interpretability is particularly valuable in mental health applications, where transparency and human oversight are essential.

In this work, we present the Explain–Retrieve–Verify (ERV) workflow, a LLM-based workflow for Arabic mental health question and answer classification, developed for the AraHealthQA 2025 shared task. ERV operates in three steps: (i) the *Explain step*, where an LLM generates candidate labels with rationales; (ii) the *Retrieve step*, where semantically similar training examples are surfaced to provide case-based evidence; and (iii) the *Verify step*, where LLM reconciles both sources to produce final labels with calibrated confidence. On the official test set, ERV achieves a weighted F1-score of 0.61 and Jaccard score of 0.53 for question classification, outperforming fine-tuned baselines, and when combined with MARBERT (Abdul-Mageed et al., 2020) reaches 0.80 weighted F1 and 0.72 Jaccard for answer classification, the best overall performance.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 226–232
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Task Definition

The AraHealthQA 2025 shared task (Alhuzali et al., 2025) focuses on Arabic health question-answering with two primary tracks. Track 1 (MentalQA) addresses classification of mental health questions and answers, while Track 2 handles general health topics. We participate in Track 1 sub-track 1 and 2: question classification and answer classification. Both sub-tasks are multi-label classification tasks where instances can belong to multiple categories simultaneously. The competition uses the MentalQA dataset (Alhuzali et al., 2024) with a train/dev/test split of 300/50/150 samples. Evaluation employs weighted F1-score and Jaccard index as primary metrics.

3 Data

The MentalQA dataset contains 500 Arabic question-answer pairs collected from Altibbi.com, focusing on mental health interactions posted between 2020-2021. The dataset encompasses interactions between patients seeking mental health guidance and professional doctors providing responses. Questions are classified into seven types: (A) Diagnosis, (B) Treatment, (C) Anatomy & Physiology, (D) Epidemiology, (E) Healthy Lifestyle, (F) Provider Choice, and (Z) Other. For detailed category definitions and examples, see (Alhuzali et al., 2024). Doctor responses are classified into three communication strategies: (1) Information, (2) Direct Guidance, and (3) Emotional Support. Complete strategy descriptions are available in the original dataset paper (Alhuzali et al., 2024).

4 Method

We present Explain–Retrieve–Verify (ERV) workflow, a training-free multi-agent pipeline that combines explicit reasoning, similarity-based retrieval, and consensus verification for Arabic medical text classification. The system operates through three sequential agents that provide complementary perspectives on multi-label classification decisions.

Explain The Explain step sends either the question or the answer to an LLM with chain-of-thought prompting with Arabic medical contexts. The agent outputs predicted labels, explanations, and confidence scores about the LLM own answer.

Retrieve The Retrieve step identifies semantically similar training examples through embedding-based similarity search. We pre-encode all training texts and cache these embeddings. For each input, we retrieve the k-nearest neighbors based on cosine similarity, and analyze their label patterns. The step passes the retrieved examples to an LLM asking to perform pattern analysis to suggest appropriate categories based on the given similar training cases

4.1 Verify

The Verify step reconciles the outputs from the Explain and Retrieve steps. We prompt LLM with three inputs: (i) the label predictions, rationales, and confidence score from the Explain step, (ii) the retrieved examples with their gold labels and the suggested categories from the Retrieve step, and (iii) the full task taxonomy. The verifier is instructed to compare the two sources of evidence, identify agreements and conflicts, and produce a final multi-label decision. It outputs the final label set, a calibrated confidence score, and a short reconciliation note explaining how disagreements were resolved.

5 Experimental Setup

For fine-tuned baselines, we trained MARBERT (Abdul-Mageed et al., 2020) and AraBERT-v02 (Antoun et al., 2020) using standard hyperparameters: learning rate 2×10^{-5}, batch size 16, 5 epochs with the best checkpoint selected by the highest weighted F1 on the validation set, and weight decay 0.01. For the ERV pipeline, we used GPT-4 as the underlying language model and employed a multilingual Sentence-BERT model (Reimers and Gurevych, 2019)[1] to compute semantic embeddings in the *Retrieve* step. The entire ERV system was implemented using the DSPy framework[2], which provides modular components for prompt engineering and multi-step reasoning workflows. All experiments were conducted on a single NVIDIA A100 GPU on Google Colab. The code used for the experiments is available on GitHub[3].

6 Results

We evaluate our ERV pipeline against fine-tuned Arabic language models on both question and answer classification sub-tasks using weighted F1-

[1] paraphrase-multilingual-mpnet-base-v2, https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

[2] https://dspy.ai/

[3] https://github.com/AhmedAbdel-Aal/mucAI-at-AraHealthQA-2025

Method	Weighted F1	Jaccard
MARBERT	0.56	0.51
AraBERT	0.56	0.51
ERV	**0.61**	**0.53**

Table 1: Question Classification Results on the test set.

Method	Weighted F1	Jaccard
MARBERT	0.76	**0.73**
AraBERT	0.74	0.68
ERV	0.73	0.61
MARBERT + ERV	**0.80**	0.72

Table 2: Answer Classification Results on the test set.

score and instance-based Jaccard index as primary metrics.

Table 1 shows the performance comparison for question classification on the test set. ERV achieves the best performance with a weighted F1-score of 0.61 and instance-based Jaccard of 0.53, outperforming both fine-tuned baselines. The improvement demonstrates the effectiveness of the multi-agent collaboration approach over single-model fine-tuning.

For answer classification (Table 2), fine-tuned models show stronger performance, with MARBERT achieving 0.76 weighted F1-score. ERV performs competitively at 0.73 weighted F1-score but falls behind the fine-tuned baselines on this subtask. Observing that MARBERT struggled to predict label 3 (Emotional Support) during development, we designed a hybrid approach combining MARBERT's expertise with ERV's pattern recognition capabilities. In this hybrid system, MARBERT serves as the primary classifier while ERV specifically handles the detection of emotional support responses. This combination achieves the best overall performance with 0.80 weighted F1-score and 0.72 instance-based Jaccard.

7 Discussion

7.1 Per Label Analysis

Comparing ERV directly with MARBERT reveals fundamentally different approaches to handling class imbalance and provides insights into why each method excels in different scenarios. Tables 3 and 4 present the detailed per-class performance comparison.

The comparison reveals fundamentally different precision-recall strategies between the two approaches. ERV consistently achieves higher recall across most question categories, particularly for dominant classes (A: 0.94 vs 0.79, B: 0.86 vs 0.74). For answers, ERV demonstrates an extreme high-recall strategy on Strategy 2 (Direct Guidance: 0.99 vs 0.86), while MARBERT shows higher recall on Strategy 1(0.89 vs 0.69).

The most striking difference lies in minority class handling. ERV shows remarkable ability to detect minority classes that MARBERT completely misses. For questions, ERV achieves non-zero performance on categories C (F1: 0.20) and F (F1: 0.15), while MARBERT scores zero on these categories. For answers, ERV detects emotional support responses (Strategy 3) with substantial performance (F1: 0.44, precision: 0.37, recall: 0.56), while MARBERT achieves zero performance. This represents a critical 44-point F1 advantage for detecting emotional support in mental health contexts. We hypothesize that this capability stems from ERV's similarity-based approach, which can identify minority class instances by matching them to semantically similar training examples. The hybrid approach validates this complementary strength, achieving the best overall performance (weighted F1: 0.80) by combining MARBERT's precision on majority classes with ERV's minority class detection capabilities.

7.2 Interpretability

A key advantage of large language models is their ability to verbalize reasoning, providing a level of transparency not available in neural fine-tuned models. Our ERV pipeline makes this explicit by combining rationales from the classification agent, evidence from retrieved examples, and the reconciliation notes from the verifier. This interpretability is particularly valuable in mental-health contexts, where system outputs should not only predict labels but also justify decisions in a way that is accessible to human reviewers. To illustrate, we show in Figure 1 a full example of the ERV workflow for question classification, and in Figure 2 an example of the ERV workflow for answer classification.

8 Limitations and Future Work

The current evaluation is constrained by testing single representatives of each modeling paradigm (MARBERT/AraBERT for fine-tuning, multilingual-mpnet-base for Encoding, and GPT-4 for ERV), which limits the generalizability of

Category	**ERV**			**MARBERT**			**Support**
	P	**R**	**F1**	**P**	**R**	**F1**	
A (Diagnosis)	0.67	0.94	0.78	0.68	0.79	0.73	84
B (Treatment)	0.63	0.86	0.73	0.72	0.74	0.73	85
C (Anatomy)	0.20	0.20	0.20	0.00	0.00	0.00	10
D (Epidemiology)	0.35	0.21	0.26	0.47	0.21	0.29	34
E (Lifestyle)	0.38	0.61	0.47	0.41	0.29	0.34	38
F (Provider Choice)	0.14	0.17	0.15	0.00	0.00	0.00	6
Z (Other)	0.00	0.00	0.00	0.00	0.00	0.00	3
Weighted Avg	0.54	0.71	0.61	0.57	0.57	0.56	260

Table 3: Per-class performance comparison for question classification

Strategy	**ERV**			**MARBERT**			**Hybrid**	**Support**
	P	**R**	**F1**	**P**	**R**	**F1**	**F1**	
1 (Information)	0.85	0.69	0.76	0.84	0.89	0.87	0.87	112
2 (Direct Guidance)	0.60	0.99	0.75	0.74	0.86	0.80	0.80	86
3 (Emotional Support)	0.37	0.56	0.44	0.00	0.00	0.00	0.44	18
Weighted Avg	0.71	0.80	0.73	0.73	0.81	0.77	0.80	216

Table 4: Per-class performance comparison for answer classification with hybrid results

our comparative findings. Future work should expand the experimental scope to include diverse Arabic language models of varying sizes and capabilities to establish more robust conclusions about the trade-offs between approaches.

While ERV demonstrates promising results, several limitations suggest important directions for future research. ERV requires multiple LLM calls per instance (three sequential steps plus similarity computation), resulting in significantly higher computational costs compared to single forward passes in fine-tuned models.

Our hypothesis that ERV's similarity-based retrieval drives its minority class detection capability suggests an intriguing research direction: interpolating fine-tuned models like MARBERT with k-nearest neighbors (kNN) at inference time. This approach could potentially provide MARBERT with non-zero performance on minority classes by incorporating retrieval-based evidence while maintaining its strong performance on majority categories.

For PLMs, performance is bounded by the small size of the used dataset. Future work can explore curating a larger corpus with balanced label coverage, and low-risk augmentation (back-translation, controlled paraphrasing).

The interpretable nature of ERV's three-step workflow creates opportunities for human-in-the-loop systems where medical professionals can review and refine step-by-step reasoning. Additionally, more sophisticated verification mechanisms could incorporate uncertainty quantification, confidence-aware voting, and learned arbitration strategies beyond the current simple consensus approach.

9 Conclusion

We present ERV (Explain–Retrieve–Verify), a three-step workflow for Arabic mental health question and answer classification. Our approach combines three sequential steps: the Explain step provides initial predictions through chain-of-thought reasoning, the Retrieve step identifies similar examples for evidence-based analysis, and the Verify step reconciles both signals to produce final classifications with calibrated confidence. Our experiments show that the ERV workflow improves over fine-tuned language models on the question classification task and provides complementary strengths for answer classification, especially in detecting emotional support strategies. Our work contributes to Arabic mental health NLP by demonstrating that collaborative three-step reasoning workflows can compete with fine-tuned models while offering better interpretability.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert &

marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Mouath Abu Daoud, Chaimae Abouzahir, Leen Kharouf, Walid Al-Eisawi, Nizar Habash, and Farah E Shamout. 2025. Medarabiq: Benchmarking large language models on arabic medical tasks. *arXiv e-prints*, pages arXiv–2505.

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash, and Leen Kharouf. 2025. Arahealthqa 2025 shared task description paper. In *Proceedings of ArabicNLP 2025*.

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.

Thushari Atapattu, Mahen Herath, Charitha Elvitigala, Piyanjali de Zoysa, Kasun Gunawardana, Menasha Thilakaratne, Kasun De Zoysa, and Katrina Falkner. 2022. Emoment: An emotion annotated mental health corpus from two south asian countries. *arXiv preprint arXiv:2208.08486*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jaya Chaturvedi, Sumithra Velupillai, Robert Stewart, and Angus Roberts. 2023. Identifying mentions of pain in mental health records text: a natural language processing approach. *arXiv preprint arXiv:2304.01240*.

Soumitra Ghosh, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Cease, a corpus of emotion annotated suicide notes in english. In *Proceedings of the twelfth language resources and evaluation conference*, pages 1618–1626.

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2021. Arabic natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences*, 33(5):497–507.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

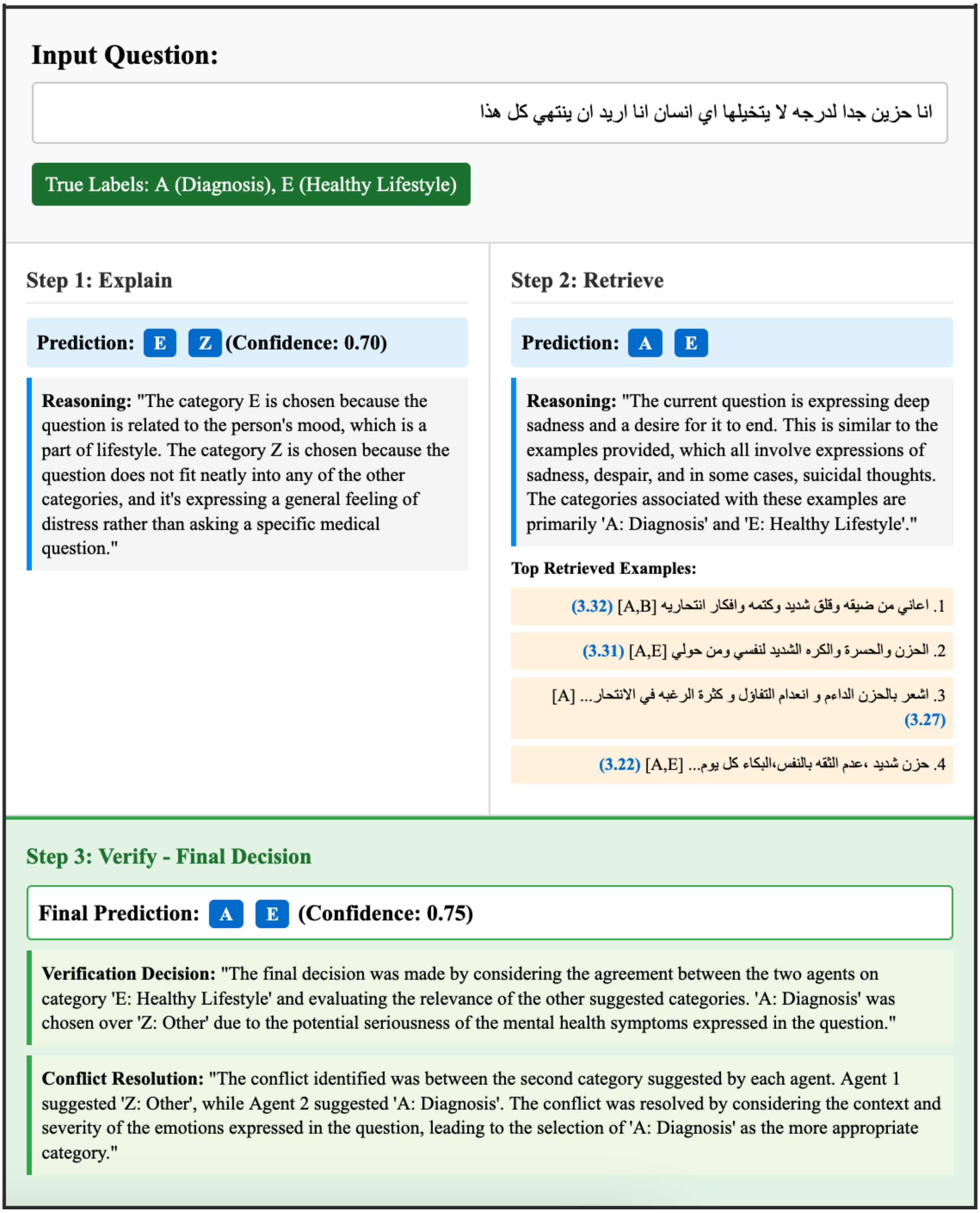
Akshay Swaminathan, Iván López, Rafael Antonio Garcia Mar, Tyler Heist, Tom McClintock, Kaitlin Caoili, Madeline Grace, Matthew Rubashkin, Michael N Boggs, Jonathan H Chen, and 1 others. 2023. Natural language processing system for rapid detection and intervention of mental health crisis chat messages. *NPJ digital medicine*, 6(1):213.

Qianqian Xie, Qingyu Chen, Aokun Chen, Cheng Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Keloth, and 1 others. 2025. Medical foundation large language models for comprehensive text analysis and beyond. *npj Digital Medicine*, 8(1):141.

Kailai Yang, Shaoxiong Ji, Tianlin Zhang, Qianqian Xie, Ziyan Kuang, and Sophia Ananiadou. 2023. Towards interpretable mental health analysis with large language models. *arXiv preprint arXiv:2304.03347*.

Monica Zolezzi, Maha Alamri, Shahd Shaar, and Daniel Rainkie. 2018. Stigma associated with mental illness and its treatment in the arab culture: a systematic review. *International Journal of Social Psychiatry*, 64(6):597–609.

A Appendix A



Input Question:

انا حزين جدا لدرجه لا يتخيلها اي انسان انا اريد ان ينتهي كل هذا

True Labels: A (Diagnosis), E (Healthy Lifestyle)

Step 1: Explain

Prediction: E Z **(Confidence: 0.70)**

Reasoning: "The category E is chosen because the question is related to the person's mood, which is a part of lifestyle. The category Z is chosen because the question does not fit neatly into any of the other categories, and it's expressing a general feeling of distress rather than asking a specific medical question."

Step 2: Retrieve

Prediction: A E

Reasoning: "The current question is expressing deep sadness and a desire for it to end. This is similar to the examples provided, which all involve expressions of sadness, despair, and in some cases, suicidal thoughts. The categories associated with these examples are primarily 'A: Diagnosis' and 'E: Healthy Lifestyle'."

Top Retrieved Examples:

1. اعاني من ضيقه وقلق شديد وكتمه وافكار انتحاريه [A,B] (3.32)
2. الحزن والحسرة والكره الشديد لنفسي ومن حولي [A,E] (3.31)
3. اشعر بالحزن الداءم و انعدام التفاؤل و كثرة الرغبه في الانتحار... [A] (3.27)
4. حزن شديد ،عدم الثقه بالنفس،البكاء كل يوم... [A,E] (3.22)

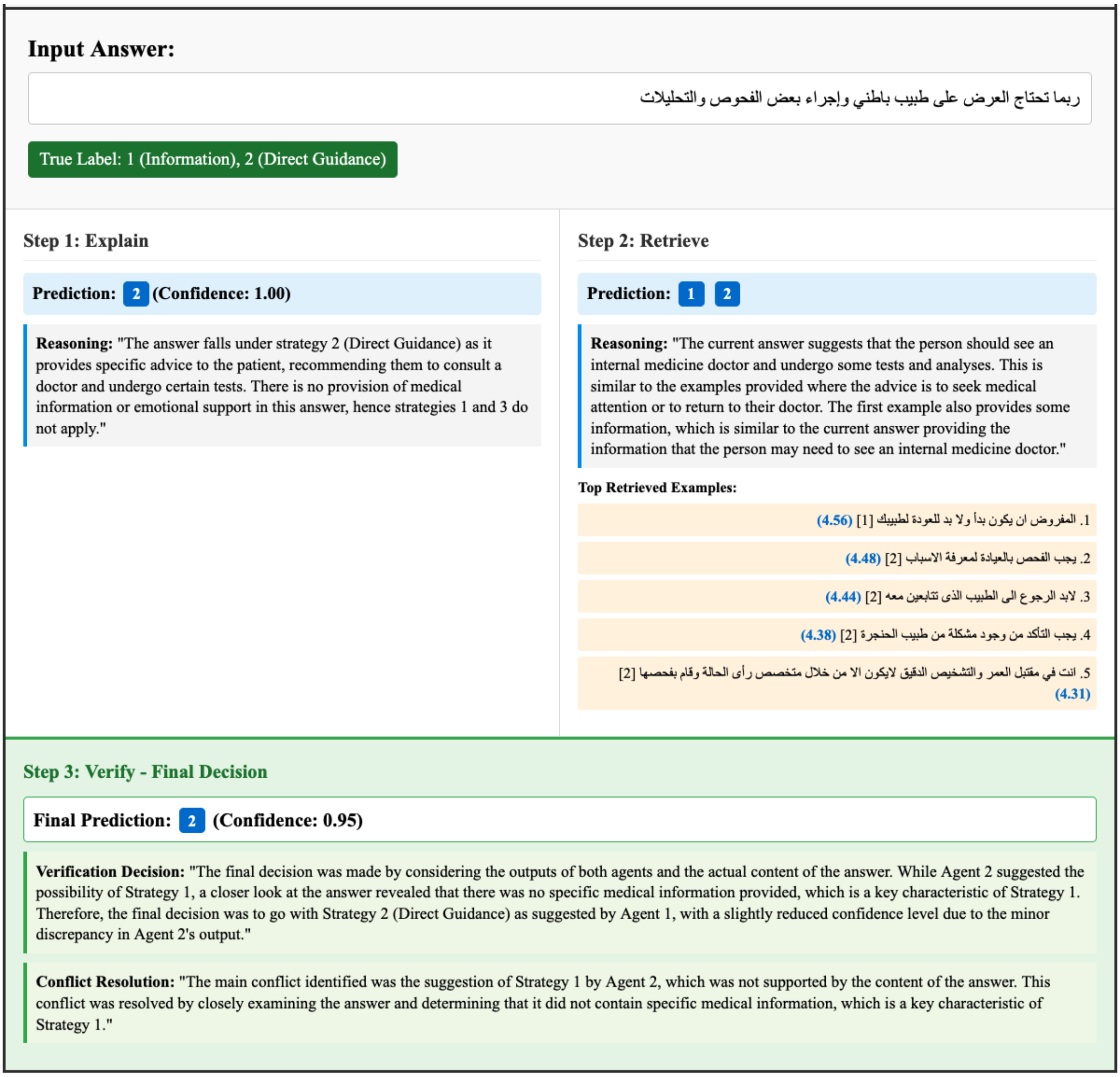
Step 3: Verify - Final Decision

Final Prediction: A E **(Confidence: 0.75)**

Verification Decision: "The final decision was made by considering the agreement between the two agents on category 'E: Healthy Lifestyle' and evaluating the relevance of the other suggested categories. 'A: Diagnosis' was chosen over 'Z: Other' due to the potential seriousness of the mental health symptoms expressed in the question."

Conflict Resolution: "The conflict identified was between the second category suggested by each agent. Agent 1 suggested 'Z: Other', while Agent 2 suggested 'A: Diagnosis'. The conflict was resolved by considering the context and severity of the emotions expressed in the question, leading to the selection of 'A: Diagnosis' as the more appropriate category."

Figure 1: ERV pipeline workflow showing the three-step process for Arabic mental health question classification. The Explain step provides initial predictions through chain-of-thought reasoning, the Retrieve step identifies similar examples for pattern analysis, and the Verify step reconciles predictions to produce the final classification of A (Diagnosis) and E (Healthy Lifestyle).



Input Answer:

ربما تحتاج العرض على طبيب باطني وإجراء بعض الفحوص والتحليلات

True Label: 1 (Information), 2 (Direct Guidance)

Step 1: Explain

Prediction: 2 **(Confidence: 1.00)**

Reasoning: "The answer falls under strategy 2 (Direct Guidance) as it provides specific advice to the patient, recommending them to consult a doctor and undergo certain tests. There is no provision of medical information or emotional support in this answer, hence strategies 1 and 3 do not apply."

Step 2: Retrieve

Prediction: 1 2

Reasoning: "The current answer suggests that the person should see an internal medicine doctor and undergo some tests and analyses. This is similar to the examples provided where the advice is to seek medical attention or to return to their doctor. The first example also provides some information, which is similar to the current answer providing the information that the person may need to see an internal medicine doctor."

Top Retrieved Examples:

1. المفروض ان يكون بدأ ولا بد للعودة لطبيبك [1] **(4.56)**
2. يجب الفحص بالعيادة لمعرفة الاسباب [2] **(4.48)**
3. لابد الرجوع الى الطبيب الذى تتابعين معه [2] **(4.44)**
4. يجب التأكد من وجود مشكلة من طبيب الحنجرة [2] **(4.38)**
5. انت في مقتبل العمر والتشخيص الدقيق لايكون الا من خلال متخصص رأى الحالة وقام بفحصها [2] **(4.31)**

Step 3: Verify - Final Decision

Final Prediction: 2 **(Confidence: 0.95)**

Verification Decision: "The final decision was made by considering the outputs of both agents and the actual content of the answer. While Agent 2 suggested the possibility of Strategy 1, a closer look at the answer revealed that there was no specific medical information provided, which is a key characteristic of Strategy 1. Therefore, the final decision was to go with Strategy 2 (Direct Guidance) as suggested by Agent 1, with a slightly reduced confidence level due to the minor discrepancy in Agent 2's output."

Conflict Resolution: "The main conflict identified was the suggestion of Strategy 1 by Agent 2, which was not supported by the content of the answer. This conflict was resolved by closely examining the answer and determining that it did not contain specific medical information, which is a key characteristic of Strategy 1."

Figure 2: ERV workflow for answer strategy classification. The Explain step identifies direct guidance, the Retrieve step finds mixed evidence from similar examples, and the Verify step resolves the conflict by analyzing the absence of specific medical information, correctly classifying the answer as strategy 2 (Direct Guidance) while missing the 1 (Information) Strategy.

MarsadLab at AraHealthQA: Hybrid Contextual–Lexical Fusion with AraBERT for Question and Answer Categorization

Mabrouka Bessghaier[2], Shimaa Amer Ibrahim [2], Md. Rafiul Biswas[1], Wajdi Zaghouani[2]
[1]Hamad Bin Khalifa University, Qatar, [2]Northwestern University in Qatar, Qatar
mbiswas@hbku.edu.qa
{mabrouka.bessghaier,shimaa.ibrahim,wajdi.zaghouani}@northwestern.edu

Abstract

This paper presents the MarsadLab submission to Track 1 of the AraHealthQA 2025 Shared Task, addressing two subtasks: (A) multi-label question categorization and (B) multi-label answer categorization in Arabic mental health discourse. Our approach employs a hybrid contextual–lexical fusion architecture built on AraBERTv2, enriched with task-specific handcrafted features such as lexical indicators, linguistic cues, and domain-informed keyword signals. On the official test set, the system achieved a weighted F1 score of 0.55 (Jaccard 0.41) for Task A and 0.79 (Jaccard 0.67) for Task B.

1 Introduction

Mental health strongly shapes how people think, feel, and function, and untreated conditions such as anxiety, depression, or cognitive disorders can severely reduce quality of life. This growing societal need has motivated the use of computational methods to support mental health understanding and intervention.

Meanwhile, advances in Large Language Models (LLMs) such as GPT-3 (Brown et al., 2020), LAMDA (Thoppilan et al., 2022), PaLM (Chowdhery et al., 2023), and LLaMA (Touvron et al., 2023) have transformed NLP and shown promise in healthcare applications (Sakai and Lam, 2025). Yet their potential remains underexplored in Arabic, particularly for mental health.

Research on Arabic mental health NLP is still in its early stages. For instance, (Mezzi et al., 2022) used BERT-based intent recognition (Devlin et al., 2019) with the MINI framework to diagnose conditions such as depression, suicidality, and panic disorder, achieving nearly 90% accuracy. Nevertheless, benchmarks and resources remain scarce, highlighting the need for community-driven initiatives in this area.

2 Background

The AraHealthQA 2025 Shared Task (Alhuzali et al., 2025) introduces the first benchmark for Arabic medical question answering, with two tracks: Mental Health QA (MentalQA) and General Health QA (MedArabiQ). Our work focuses on MentalQA, specifically **Subtask A: Question Categorization** and **Subtask B: Answer Categorization**. In Task A, the system classifies questions into categories such as Diagnosis or Epidemiology. For example:

هل يعتبر الخوف من عدم الإنجاب مستقبلاً
حالة عادية خاصةً لما أكون متعلقة بأطفال كثيراً
وأنا على وجه جواز أنا خايفة جداً
(Is the fear of not being able to have children in the future normal, especially since I am very attached to children and about to get married?)

The expected output is **A (Diagnosis)** and **D (Epidemiology)**. In Task B, the system instead classifies answers by strategy, such as **1 (Information)** and **2 (Direct Guidance)** for the same example.

The dataset for Track 1 is a newly introduced resource comprising approximately 500 manually annotated Arabic question–answer pairs in the mental health domain (Alhuzali et al., 2024). The data is primarily in Modern Standard Arabic with some dialectal variation, collected from user-generated online content. Each instance may carry multiple overlapping labels, reflecting the complexity of real mental health communication.

Recent efforts in Arabic health-related NLP and conversational AI highlight both the opportunities

Proceedings of The Third Arabic Natural Language Processing Conference, pages 233–238
November 8-9, 2025 ©2025 Association for Computational Linguistics

and challenges for building robust health QA systems. Prior research on Arabic chatbots emphasizes the design of dialog systems for clinical intents, leveraging techniques such as intent classification, NER, and slot filling, while also noting gaps in evaluation protocols, resources, and ethical considerations such as privacy and bias (Ahmed et al., 2022). Parallelly, the fight against health misinformation, particularly during COVID-19, has driven the development of annotation frameworks, credibility signals, and check-worthiness pipelines across multiple languages (Alam et al., 2021; Nakov et al., 2022), providing valuable methodologies for grounding QA in trustworthy evidence. On the mental health side, studies analyzing Arabic social media discourse, such as depression expression (Mohamed and Zaghouani, 2024) and COVID-19-related loneliness (Shurafa and Zaghouani, 2025), demonstrate the feasibility of corpus-driven modeling of emotional and psychological signals, while underscoring ethical concerns around data sensitivity. More recent research has focused on the intersection of Arabic NLP and mental health, including comprehensive surveys of methods and resources (Alasmari, 2025), empirical evaluations of pre-trained language models for Arabic Q/A classification in mental health (Alhuzali and Alasmari, 2025), and applied systems such as the bilingual MindWave app for AI-driven support (Bensalah et al., 2024). Additionally, large-scale evaluations of LLMs in the Arabic mental health domain (Zahran et al., 2025) shed light on both the promise and limitations of current models. The AraHealthQA 2025 shared task is situated within this growing body of work, aiming to foster resources and benchmarks for Arabic mental health and medical QA.

Our contribution lies in integrating transformer-based contextual embeddings with carefully designed task-specific features, tailored to capture the linguistic and psychological nuances of Arabic mental health discourse. Specifically, we employ a hybrid contextual–lexical fusion approach that integrates AraBERTv2 representations with handcrafted lexicon and keyword features. The lexicons are automatically derived from the training data, capturing frequent tokens associated with each category, while the keyword lists are manually curated to reflect pragmatic markers of diagnosis, treatment, guidance, and emotional support. This design allows the model not only to benefit from deep contextual semantics but also to leverage interpretable and domain-relevant signals that are particularly valuable under the low-resource conditions of this shared task.

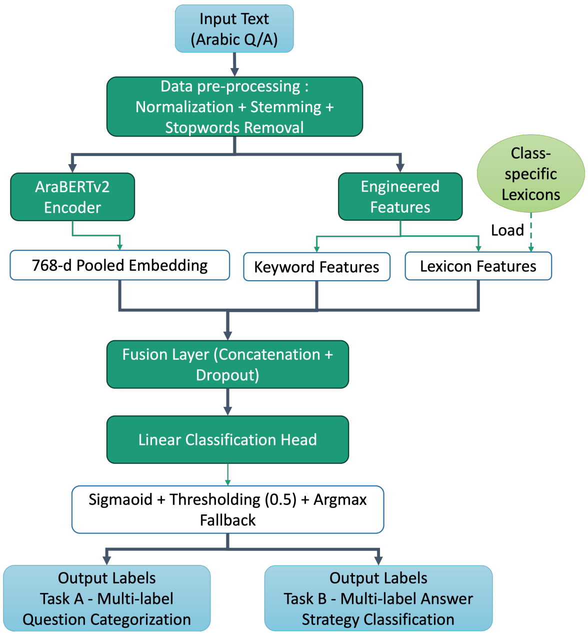


Figure 1: Hybrid Classification Architecture (instantiated separately for Task A and Task B)

3 System Overview

Our approach to both Task A and Task B is based on a hybrid architecture that integrates deep contextual embeddings with handcrafted features. The pipeline consists of three main components: (i) contextual embeddings obtained from AraBERTv2, (ii) handcrafted features capturing lexical, linguistic, and pragmatic information, and (iii) a fusion and classification layer that combines both representations.

As illustrated in Figure 1, the model follows a dual-branch design: AraBERTv2 encodes contextual semantics, while a parallel handcrafted feature block encodes lexical, linguistic, and pragmatic indicators. The two representations are concatenated, regularized with dropout, and passed through a linear classification head. Outputs are produced via sigmoid activation with a 0.5 threshold and an argmax fallback to ensure at least one label. This hybrid pipeline is applied across both tasks, with the same backbone architecture but task-specific lexicons and keyword lists tailored to Question Categorization (Task A) and Answer Strategy Classification (Task B).

3.1 Preprocessing

All input text was normalized (removing diacritics, unifying variants of alif, ya, and taa marbuta), tokenized, cleaned of non-Arabic characters and stopwords, and stemmed with the ISRI stemmer[1] to reduce words to their roots. This preprocessing ensured consistent lexical representations and allowed inflected forms to be collapsed into a single token.

3.2 Features Extraction

We incorporated two types of handcrafted features in parallel to AraBERT embeddings:

Lexicon Features. For each label set, we built lexicons by extracting the top 40 most frequent non-stopword tokens from the preprocessed training data. During feature extraction, these lexicons were used as lookup tables. For each input and each class, we checked whether at least one token from the class lexicon appeared in the text: if true, the feature value was set to 1; otherwise, it was 0. Importantly, this means that multiple overlaps do not increase the score—the feature encodes only the binary presence or absence of a lexicon match. The resulting binary vector was then concatenated with other features and AraBERT embeddings.

Keyword Features. We defined manually curated keyword lists to capture domain-relevant and pragmatic expressions (e.g., definitional markers, directive verbs, supportive phrases). These keywords were stemmed and matched in the input text, with binary features assigned to indicate their presence.

Both lexicon-based and keyword-based features were concatenated with AraBERT embeddings, enabling the model to exploit not only contextual semantics but also interpretable lexical and pragmatic cues.

3.3 Sub-Task A: Multi-label Question Categorization

Step 1: Contextual Embeddings. Each question was encoded with AraBERTv2[2], producing a 768-dimensional pooled embedding.

Step 2: Features Extraction. For Task A, we built class-specific lexicons by extracting the top 40 most frequent non-stopword tokens from the preprocessed, stemmed training questions associated with each label. At inference, The created lexicon is used as a lookup to compute a binary presence feature per class. In parallel, curated keyword lists were defined based on question categories, such as whether the question seeks a diagnosis, treatment, or lifestyle advice. Both lexicon and keyword indicators were concatenated with AraBERT embeddings to enrich the representation of each question. Representative examples are shown in Table 1, with the full lists in Appendix A.

Category	Keywords Examples
A	أعراض (Symptoms)
B	علاج (Treatment), دواء (Medicine)
C	جسم (Body), دماغ (Brain)
D	سبب (Cause), عوامل (Factors)
E	رياضة (Exercise), نوم (Sleep)
F	طبيب (Doctor), مستشفى (Hospital)

Table 1: Example keywords associated with each label category (Task A)

Answer Strategy	Example Keywords
1 : Information	معلومة, تشخيص, اعراض
2 : Direct Guidance	أنصح, يجب, نصيحة
3 : Emotional Support	معك, تقلق, اطمئن

Table 2: Examples of defined keywords for Answer classification (Task B)

Step 3: Fusion and Classification. The AraBERT embeddings and handcrafted features were concatenated, regularized with dropout, and passed through a linear layer. Predictions were obtained via sigmoid activation with a 0.5 threshold and argmax fallback. Besides, category Z (Other) acts as a default class whenever a question does not strongly align with any of the six primary categories (A–F).

3.4 Sub-Task B: Multi-label Answer Strategy Classification

Step 1: Contextual Embeddings. Each answer was encoded with AraBERTv2, producing a 768-dimensional pooled embedding.

Step 2: Features Extraction. For Task B, we built lexicons for each of the three answer strategies (Information, Direct Guidance, Emotional Support)

[1] https://www.nltk.org/_modules/nltk/stem/isri.html

[2] aubmindlab/bert-base-arabertv2

by extracting the top 40 tokens from the training data for each class. As with Task A, lexicon features were computed as binary indicators. Additionally, curated keyword lists were integrated, which capture pragmatic signals such as definitional markers, directive verbs, and supportive expressions. Representative examples are shown in Table 2, with full lists in Appendix B.

Step 3: Fusion and Classification. As in Task A, embeddings and handcrafted features were concatenated, passed through dropout, and classified using a linear layer followed by sigmoid activation with thresholding and argmax fallback.

4 Experimental Setup

Following the AraHealth shared task, we evaluate using weighted F1-score and Jaccard similarity. The model is optimized with binary cross-entropy loss and AdamW (learning rate 1.5). A sigmoid threshold of 0.5 is used to convert probabilities into binary predictions, and an argmax fallback ensures that at least one label is always assigned. Early stopping with patience (3–5 epochs) is applied based on validation loss to prevent overfitting.

5 Results and Discussion

5.1 Main Findings

Task A. The model demonstrates strong performance on categories with salient lexical cues such as Diagnosis and Treatment, while categories characterized by diffuse semantics including Epidemiology and Other present greater classification challenges. Our approach achieves a weighted F1-score of 0.55 and weighted Jaccard similarity of 0.41 on the official test set.

Task B. The classification hierarchy reveals that Information detection yields the highest accuracy, followed by Direct Guidance identification. Notably, Emotional Support frequently exhibits confusion with guidance categories due to overlapping pragmatic markers. The model attains superior performance on the official test set with a weighted F1-score of 0.79 and weighted Jaccard similarity of 0.67.

5.2 Discussion

We conducted comprehensive ablation studies to quantify the contribution of different feature combinations: AraBERT-only baseline, AraBERT+Lexicon features, AraBERT+Keywords, and the AraBERT+Lexicon+Keywords combination. Table 3 presents the detailed results. In addition, we also explored using lexicon and keyword features with a traditional classifier such as SVM to examine their standalone effectiveness outside the AraBERT architecture.

Task A Performance Analysis: The AraBERT baseline achieved F1=0.52 and Jaccard=0.39, with keyword features yielding the strongest gains (F1=0.56, Jaccard=0.43). Lexicon features alone slightly reduced performance, while the combined setup offered balanced improvements (F1=0.55, Jaccard=0.41).To establish a comparative baseline beyond transformer architectures, we implemented a traditional Support Vector Machine utilizing lexicon and keyword features exclusively. This classical approach demonstrated competitive performance with F1=0.45 and Jaccard=0.34, representing a notable accomplishment without the computational overhead of large language models, though performance remained consistently below all AraBERT configurations.

Task B Performance Analysis: The AraBERT baseline delivered strong performance (F1=0.79, Jaccard=0.69), with feature integration yielding minimal changes. Lexicon features slightly reduced performance, while keywords and combined features maintained near-baseline results. The SVM implementation achieved competitive performance with F1=0.74 and Jaccard=0.62, demonstrating effective classification capabilities.

The experimental findings indicate that lexicon features demonstrate optimal effectiveness for categories characterized by stable, domain-specific terminology (e.g., Diagnosis, Treatment), while keyword features exhibit particular strength in supporting pragmatic classification tasks (Direct Guidance, Emotional Support). Notably, Task A reveals that keyword features individually outperform other configurations, suggesting their superior utility for question classification compared to lexicon-based approaches. However, the dataset suffers from severe class imbalance—especially in Task A—which substantially affects overall model performance.

6 Conclusion

We proposed a hybrid architecture that integrates AraBERTv2 contextual embeddings with a compact set of handcrafted features, including lexical indicators, linguistic cues, and domain-informed

Task	Variant	F1	Jaccard
A	AraBERT only	0.52	0.39
	+ Lexicon	0.51	0.37
	+ Keywords	0.56	0.43
	+ Lexicon + Keywords	0.55	0.41
	SVM + features	0.45	0.34
B	AraBERT only	0.79	0.69
	+ Lexicon	0.77	0.67
	+ Keywords	0.78	0.68
	+ Lexicon + Keywords	0.79	0.67
	SVM + features	0.74	0.62

Table 3: Ablation study results on official test set.

keywords. Our findings demonstrate that fusing transformer-based representations with carefully engineered lexical and pragmatic features yields robust performance in both question categorization and answer strategy classification, even on small-scale, domain-specific datasets.

Limitations

Lexicon and keyword features were effective with SVM, but did not improve AraBERT, suggesting the limitation lies in the fusion strategy rather than the features themselves. Another issue is the restricted coverage of the manually defined keywords, which work well for dominant categories but miss diverse expressions; automated expansion or domain-specific terminologies could help. Finally, the dataset size and imbalance remain major challenges: Task A suffers from severe class imbalance with poor recall on minority labels, while Task B shows moderate imbalance, both limiting generalization.

Acknowledgements

This study was supported by the grant NPRP14C-0916-210015, awarded by the Qatar Research, Development and Innovation Council (QRDI).

References

Arfan Ahmed, Nashva Ali, Mahmood Alzubaidi, Wajdi Zaghouani, Alaa Abd-alrazaq, and Mowafa Househ. 2022. Arabic chatbot technologies: A scoping review. *Computer Methods and Programs in Biomedicine Update*, 2:100057.

Firoj Alam, Shaden Shaar, Fahim Dalvi, Hassan Sajjad, Alex Nikolov, Hamdy Mubarak, Giovanni Da San Martino, Ahmed Abdelali, Nadir Durrani, Kareem Darwish, Abdulaziz Al-Homaid, Wajdi Zaghouani, Tommaso Caselli, Gijs Danoe, Friso Stolk, Britt Bruntink, and Preslav Nakov. 2021. Fighting the COVID-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 611–649, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A. Alasmari. 2025. A scoping review of arabic natural language processing for mental health. *Healthcare*, 13(9):963.

H. Alhuzali and A. Alasmari. 2025. Pre-trained language models for mental health: An empirical study on arabic q&a classification. *Healthcare*, 13(9):985.

Hassan Alhuzali, Ashwag Alasmari, and Hamad Alsaleh. 2024. Mentalqa: An annotated arabic corpus for questions and answers of mental healthcare. *IEEE Access*, 12:101155–101165.

Hassan Alhuzali, Farah Shamout, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Walid Al-Eisawi, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Nizar Habash, and Leen Kharouf. 2025. Arahealthqa 2025 shared task description paper. In *Proceedings of ArabicNLP 2025*.

N. Bensalah, H. Ayad, A. Adib, and A. I. El Farouk. 2024. Mindwave app: Leveraging ai for mental health support in english and arabic. In *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 1–6. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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.

Ridha Mezzi, Aymen Yahyaoui, Mohamed Wassim Krir, Wadii Boulila, and Anis Koubaa. 2022. Mental health intent recognition for arabic-speaking patients using the mini international neuropsychiatric interview (mini) and bert model. *Sensors*, 22(3):846.

Ahd Mohamed and Wajdi Zaghouani. 2024. Expression of depression among arab twitter users using arabic corpus analysis. *Procedia Computer Science*, 244:76–85. 6th International Conference on AI in Computational Linguistics.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Julia Maria Struß, Thomas Mandl, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, Chengkai Li, Shaden Shaar, Gautam Kishore Shahi, Hamdy Mubarak, Alex Nikolov, Nikolay Babulkov, Yavuz Selim Kartal, and Javier Beltrán. 2022. The clef-2022 checkthat! lab on fighting the covid-19 infodemic and fake news detection. In *Advances in Information Retrieval*, pages 416–428, Cham. Springer International Publishing.

Hajar Sakai and Sarah S Lam. 2025. Large language models for healthcare text classification: A systematic review. *arXiv preprint arXiv:2503.01159*.

Chereen Shurafa and Wajdi Zaghouani. 2025. Corpus analysis of covid-19 related loneliness on twitter. In *Arabic Language Processing: From Theory to Practice*, pages 80–93, Cham. Springer Nature Switzerland.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

N. Zahran, A. E. Fouda, R. J. Hanafy, and M. E. Fouda. 2025. A comprehensive evaluation of large language models on mental illnesses in arabic context. *arXiv preprint arXiv:2501.06859*.

A Keywords List for Task A

This appendix lists the curated keywords associated with each category label for the subtask A:

Label A: Diagnosis (questions about interpreting clinical findings)

مؤشرات, نتائج, تحليل, علامات,أعراض, تشخيص,
توصيف, تقرير, كشف فحص, اضطراب, حالة, مرض

Label B: Treatment (questions about seeking treatments)

خطة, معالج, وصفة, العلاج, جلسات, دواء, علاج,
نفسي, طبيعي, مسكن, مهدئ, مضاد, برنامج

Label C: Anatomy and Physiology (questions about basic medical knowledge)

إفراز, كيمياء, هرمون, أعصاب, عقل مخ, دماغ, جسم,
دوبامين, سيروتونين, فيسيولوجيا, بيولوجيا, خلايا,
جهاز, عضو, وظائف, تشريح, أدرينالين, كورتيزول,
آلية, بنية, تركيب

Label D: Epidemiology (questions about course, prognosis, and etiology of diseases)

انتشار, عدوى, وراثة, مخاطر, مؤثرات, عوامل, سبب,
مآل, توقع, نسبة, إحصاء, احتمال, تفشي

Label E: Healthy Lifestyle (questions related to diet, exercise, and mood control)

نظام, طعام, سهر, نوم, مشي, جري, تمارين, رياضة,
تأمل, استرخاء, صحة, تغذية, لياقة, رشاقة, وزن, حمية,
يوغا

Label F: Provider Choices (questions seeking recommendations for medical professionals and facilities)

أخصائي, اختصاصي, عيادة, مستشفى, دكتور, طبيب,
معالج, مستشار, طوارئ, استشارة, توصية, مركز

B Keywords List for Task B

This appendix lists the curated keywords associated with each category label for the subtask B

Label 1: Information (factual responses)

سبب, اعراض, دراسات, تعريف, يعني, تشير, معلومة,
يظهر, علامة, يفسر, موضح, شرح, تشخيص, بيانات,
دليل

Label 2: Direct Guidance (action-oriented responses)

عليك, ينصح, افضل, جرب, حاول, ينبغي, يجب, أنصح,
نصيحة, اجراء, سلوك, اتبع, خطة, خطوة, لازم, قم

Label 3: Emotional Support (empathy and encouragement)

تفهم, قلب, اطمئن, تقلق, الله, اشعر, وحدك, معك,
تهون, مشاعر, اهتم, يهمني, ادعمك, متفهم, احساس,
ارتاح

BAREC Shared Task 2025 on Arabic Readability Assessment

Khalid N. Elmadani,[1] Bashar Alhafni,[2] Hanada Taha-Thomure,[3] Nizar Habash[1]
[1]New York University Abu Dhabi
[2]Mohamed bin Zayed University of Artificial Intelligence
[3]Zayed University
{khalid.nabigh,nizar.habash}@nyu.edu
bashar.alhafni@mbzuai.ac.ae, hanada.thomure@zu.ac.ae

Abstract

We present the results and findings of the BAREC Shared Task 2025 on Arabic Readability Assessment, organized as part of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025). The BAREC 2025 shared task focuses on automatic readability assessment using the BAREC Corpus (Elmadani et al., 2025), addressing fine-grained classification into 19 readability levels. The shared task includes two sub-tasks: sentence-level classification and document-level classification, and three tracks: (1) Strict Track, where only the BAREC Corpus is allowed; (2) Constrained Track, restricted to the BAREC Corpus, SAMER Corpus (Alhafni et al., 2024), and SAMER Lexicon (Al Khalil et al., 2020), and (3) Open Track, allowing any external resources. A total of 22 teams from 12 countries registered for the task. Among these, 17 teams submitted system description papers. The winning team achieved 87.5 QWK on the sentence-level task and 87.4 QWK on the document-level task.[1]

1 Introduction

Readability assessment plays a crucial role in education, literacy development, and language learning by ensuring that texts align with a reader's proficiency level. Mismatched readability can lead to less understanding, retention, reading speed, and engagement (DuBay, 2004; Klare, 1963). To address this, text leveling systems have been widely adopted, particularly in early education, to provide structured and measurable progress in reading development (Allington et al., 2015; Barber and Klauda, 2020).

While readability models exist for several languages, many challenges remain, particularly in fine-grained text leveling and resource-scarce languages. Systems like Fountas and Pinnell's 27-level model for English (Fountas and Pinnell, 2006) and Taha-Thomure's (2017) 19-level framework for Arabic demonstrate the importance of detailed readability classification. These fine-grained levels allow for more precise educational applications while being flexible enough to map onto coarser categories for broader applications.

RL	Grade	Example
1	KG	Majed ماجد
3	1st	The morning of Eid صباح العيد
6	2nd	جاءتني فكرة An idea came to me
10	4th	كانت رحلة ممتعة! It was an enjoyable trip!
14	8th	تعريف أصول الفقه Definition of Islamic Jurisprudence Principles
17	Uni	بين طعن القَنا وخَفْق البُنودِ Between lance thrusts and ensign flutters

Table 1: Examples by Reading Level (RL) and grade.

The Taha/Arabi21 framework (Taha-Thomure, 2017), which has been used to annotate over 9,000 children's books, plays a central role in our work. Building on this system, the **Barec** guidelines (Habash et al., 2025) offer standardized, sentence-level readability assessment across a wide range of genres and educational stages – from early childhood to postgraduate levels (see Table 1). For full guidelines in Arabic and English, we refer the reader to Habash et al. (2025).

Arabic, in particular, presents unique challenges for readability assessment due to its rich morphology, extensive lexicon, and highly ambiguous orthography. Unlike English, where well-established readability formulas and datasets exist, Arabic readability research suffers from a lack of standardized resources. This gap limits the development of robust computational models capable of accurately assessing Arabic text difficulty across different proficiency levels.

[1] https://barec.camel-lab.com/sharedtask2025

Proceedings of The Third Arabic Natural Language Processing Conference, pages 239–252
November 8-9, 2025 ©2025 Association for Computational Linguistics

The BAREC 2025 shared task is organized as part of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025), collocated with EMNLP 2025. A total of 22 teams from 12 countries registered for the shared task. Out of these, 17 teams submitted system description papers which are cited in this paper (see Table 6). This paper provides an overview of the submitted systems and presents their results.

The paper is structured as follows: §2 reviews related work. §3 outlines the sub-tasks and tracks of the shared task. §4 introduces the datasets and evaluation metrics. §5 describes the baselines and provides an overview of the submitted systems. Finally, §6 reports and discusses the results.

2 Related Work

Automatic Readability Assessment Research on automatic readability assessment has produced a wide range of datasets and resources (Collins-Thompson and Callan, 2004; Pitler and Nenkova, 2008; Feng et al., 2010; Vajjala and Meurers, 2012; Xu et al., 2015; Nadeem and Ostendorf, 2018; Vajjala and Lučić, 2018; Deutsch et al., 2020; Lee et al., 2021). In English, many early datasets were built from textbooks, since their graded structure naturally supports readability evaluation (Vajjala, 2022). Over time, however, copyright limitations and lack of digitized materials pushed researchers to explore alternative sources, such as crowdsourced readability annotations from online platforms (Vajjala and Meurers, 2012; Vajjala and Lučić, 2018), or proficiency exams based on the CEFR framework for L2 learners (Xia et al., 2016).

Arabic Readability Efforts Research on Arabic readability has explored text leveling and assessment across several frameworks (Nassiri et al., 2023). Taha-Thomure (2017) proposed a 19-level system for educators, inspired by Fountas and Pinnell (2006), focusing on children's literature. This framework targets full texts, particularly for early education, with 11 of the 19 levels covering up to grade 4, helping teachers match books to students' reading abilities. It defines ten qualitative and quantitative criteria, including text genre, abstractness, vocabulary, dialectal proximity, authenticity, book production quality, content suitability, sentence structure, illustrations, use of diacritics, and word count. The framework was adopted by the Arab Thought Foundation under its Arabi21 initiative, which leveled over 9,000 children's books.

Other approaches applied the CEFR framework (Council of Europe, 2001) to Arabic, including frequency-based word lists from the KELLY project (Kilgariff et al., 2014), manually annotated corpora such as ZAEBUC (Habash and Palfreyman, 2022) and ReadMe++ (Naous et al., 2024), and vocabulary profiling (Soliman and Familiar, 2024). El-Haj et al. (2024) introduced DARES, a dataset derived from Saudi school materials, while the SAMER project (Al Khalil et al., 2020) produced a lexicon with a five-level readability scale, enabling the creation of the first manually annotated Arabic parallel corpus for text simplification (Alhafni et al., 2024). Bashendy et al. (2024) further presented a corpus of Arabic essays annotated for organization and style traits.

Automated Arabic readability assessment has progressed from rule-based models using surface features (Al-Dawsari, 2004; Al-Khalifa and Al-Ajlan, 2010; Hazim et al., 2022) to machine learning approaches incorporating linguistic features (Forsyth, 2014; Saddiki et al., 2018), and script-specific characteristics such as OSMAN (El-Haj and Rayson, 2016). Recent work demonstrates strong performance using pre-trained language models on the SAMER corpus (Liberato et al., 2024).

3 Task Description

The BAREC Readability Assessment Shared Task focuses on developing models for fine-grained readability classification using a 19-level framework. Participants built systems to classify texts into these readability levels at both the sentence and document levels.

Sub-tasks Participants compete in one or more of the following sub-tasks.

1. **Sentence-Level Classification (Sent)**: Predict the readability level of individual sentences.
2. **Document-Level Classification (Doc)**: Predict the readability level of a document, where a document is a collection of consecutive sentences, and the hardest sentence determines the readability level of the document.

Tracks Participants compete in one or more of the following tracks, each imposing different resource constraints:

- **Strict Track (S)**: Models must be trained **exclusively** on the BAREC Corpus (Elmadani

Split	#Documents	#Sentences	#Words
Train	1,518 (79%)	54,845 (79%)	832,743 (80%)
Dev	194 (10%)	7,310 (11%)	101,364 (10%)
Test	210 (11%)	7,286 (10%)	105,265 (10%)
All	1,922 (100%)	69,441 (100%)	1,039,371 (100%)

Table 2: BAREC Corpus splits.

et al., 2025), ensuring that results are comparable based solely on this dataset.

- **Constrained Track (C)**: Models may use the BAREC Corpus, SAMER Corpus (including document, fragment, and word-level annotations) (Alhafni et al., 2024), and the SAMER Lexicon (Al Khalil et al., 2020).
- **Open Track (O)**: No restrictions on external resources, allowing the use of any publicly available data.

With two sub-tasks and three tracks, the task results in a total of **six possible combinations**. Participants are allowed to compete in multiple sub-tasks and tracks. The goal is to encourage diverse methodological approaches while providing a structured framework for evaluating readability assessment models.

4 Shared Task Datasets and Evaluation

In this section, we present the datasets used in different tracks, describe the evaluation metrics, and outline the submission guidelines given to the participants.

4.1 Dataset

BAREC Corpus The BAREC Corpus (Elmadani et al., 2025) is the main corpus used in the shared task. It consists of 1,922 documents and 69,441 sentences classified into 19 readability levels. The corpus is split into **Train (≃80%)**, **Dev (≃10%)**, and **Test (≃10%)** at the document level. Table 2 shows the corpus splits in the level of documents, sentences, and words. Table 3 shows the label distribution across splits for sentence-level and document-level tasks.

SAMER Corpus The SAMER Corpus (Alhafni et al., 2024) consists of 4,289 documents (158K words) and 20,358 fragments classified into three readability levels. We utilize the fragments made available and reported on by Liberato et al. (2024). Table 4 provides an overview of the SAMER corpus statistics, including the Train, Dev, and Test splits.

SAMER Lexicon The SAMER Lexicon (Al Khalil et al., 2020) is a 40K-lemma leveled readability lexicon for Modern Standard Arabic (MSA). The lexicon consists of 40K lemma and part-of-speech pairs annotated into five readability levels. The lexicon was manually annotated by three language professionals from different regions in the Arab world. Table 5 shows the readability statistics in the lexicon.

Blind Test Set We provide a new blind test set created for this shared task and annotated in the 19 levels of the BAREC framework to evaluate the final results. The blind test set consists of 100 documents and 3,420 sentences.

4.2 Evaluation Metrics

Following Elmadani et al. (2025), we treat the Readability Assessment task as an ordinal classification problem and evaluate systems using the following metrics:

- **Accuracy (Acc)** The proportion of cases where predictions exactly match the reference labels in the 19-level scheme (**Acc**^{19}). We also report coarse-grained variants: **Acc**^{7}, **Acc**^{5}, and **Acc**^{3}, where the 19 levels are collapsed into 7, 5, and 3 levels, respectively (see Table 3).
- **Adjacent Accuracy (±1 Acc**^{19}**)** Also referred to as off-by-1 accuracy, this metric counts predictions as correct if they are either exact matches or differ from the reference by only one level.
- **Average Distance (Dist)** Equivalent to Mean Absolute Error (MAE), it computes the average absolute difference between predicted and reference labels.

				BAREC Corpus v1 (Sentences)									
Level-3	**Level-5**	**Level-7**	**Level-19**	**All**		**Train**		**Dev**		**Test**		**Blind Test**	
1	**1**	**1**	**1-alif**	409	1%	333	1%	44	1%	32	0%	21	1%
			2-ba	437	1%	333	1%	68	1%	36	0%	21	1%
			3-jim	1,462	2%	1,139	2%	182	2%	141	2%	69	2%
			4-dal	751	1%	587	1%	78	1%	86	1%	28	1%
		2	**5-ha**	3,443	5%	2,646	5%	417	6%	380	5%	188	5%
			6-waw	1,534	2%	1,206	2%	189	3%	139	2%	47	1%
			7-zay	5,438	8%	4,152	8%	701	10%	585	8%	296	9%
	2	**3**	**8-Ha**	5,683	8%	4,529	8%	613	8%	541	7%	263	8%
			9-ta	2,023	3%	1,597	3%	236	3%	190	3%	101	3%
		4	**10-ya**	9,763	14%	7,741	14%	1,012	14%	1,010	14%	457	13%
			11-kaf	4,914	7%	4,041	7%	409	6%	464	6%	233	7%
2	**3**	**5**	**12-lam**	14,471	21%	11,318	21%	1,491	20%	1,662	23%	682	20%
			13-mim	4,039	6%	3,252	6%	349	5%	438	6%	177	5%
3	**4**	**6**	**14-nun**	10,687	15%	8,573	16%	1,072	15%	1,042	14%	596	17%
			15-sin	2,547	4%	2,016	4%	258	4%	273	4%	171	5%
	5	**7**	**16-ayn**	1,141	2%	866	2%	114	2%	161	2%	55	2%
			17-fa	480	1%	364	1%	49	1%	67	1%	15	0%
			18-sad	103	0%	67	0%	13	0%	23	0%	0	0%
			19-qaf	116	0%	85	0%	15	0%	16	0%	0	0%
Total				69,441	100%	54,845	100%	7,310	100%	7,286	100%	3,420	100%

				BAREC Corpus v1 (Documents)									
Level-3	**Level-5**	**Level-7**	**Level-19**	**All**		**Train**		**Dev**		**Test**		**Blind Test**	
1	**1**	**1**	**1-alif**	0	0%	0	0%	0	0%	0	0%	0	0%
			2-ba	0	0%	0	0%	0	0%	0	0%	0	0%
			3-jim	1	0%	1	0%	0	0%	0	0%	0	0%
			4-dal	0	0%	0	0%	0	0%	0	0%	0	0%
		2	**5-ha**	0	0%	0	0%	0	0%	0	0%	0	0%
			6-waw	0	0%	0	0%	0	0%	0	0%	0	0%
			7-zay	1	0%	1	0%	0	0%	0	0%	1	1%
	2	**3**	**8-Ha**	1	0%	1	0%	0	0%	0	0%	1	1%
			9-ta	2	0%	1	0%	0	0%	1	0%	0	0%
		4	**10-ya**	13	1%	13	1%	0	0%	0	0%	0	0%
			11-kaf	18	1%	10	1%	6	3%	2	1%	1	1%
2	**3**	**5**	**12-lam**	192	10%	148	10%	25	13%	19	9%	7	7%
			13-mim	204	11%	170	11%	14	7%	20	10%	9	9%
3	**4**	**6**	**14-nun**	623	32%	489	32%	56	29%	78	37%	24	24%
			15-sin	399	21%	317	21%	46	24%	36	17%	25	25%
	5	**7**	**16-ayn**	267	14%	207	14%	32	16%	28	13%	20	20%
			17-fa	156	8%	130	9%	9	5%	17	8%	12	12%
			18-sad	12	1%	8	1%	2	1%	2	1%	0	0%
			19-qaf	33	2%	22	1%	4	2%	7	3%	0	0%
Total				1,922	100%	1,518	100%	194	100%	210	100%	100	100%

Table 3: Sentence-level and document-level splits across BAREC readability levels.

Split	#Documents	#Fragments	#Words
Train	2,790 (65%)	14,256 (70%)	112,828 (71%)
Dev	607 (14%)	2,948 (14%)	22,075 (14%)
Test	892 (21%)	3,154 (15%)	23,161 (15%)
All	4,289 (100%)	20,358 (100%)	158,064 (100%)

Table 4: SAMER Corpus splits.

Level	Type Count
Level I	3,545 (9%)
Level II	3,221 (8%)
Level III	5,510 (14%)
Level IV	10,130 (25%)
Level V	18,281 (45%)
Total	40,687 (100%)

Table 5: SAMER Lexicon distributions

- **Quadratic Weighted Kappa (QWK)** An extension of Cohen's Kappa (Cohen, 1968; Doewes et al., 2023), this measure evaluates agreement between predicted and true labels while penalizing larger misclassifications quadratically, giving higher weight to errors farther from the reference.

We report on **QWK** as the primary metric for ranking systems. We prioritize QWK as it better captures the ordinal nature of readability levels, providing smoother, distance-sensitive penalties for misclassifications compared to the hard thresholds of accuracy-based measures. The other metrics are reported in Appendix A.

4.3 Submission Guidelines

The shared task is organized in two phases: development and testing. During the **development phase**, we set up CodaBench (Xu et al., 2022) competitions for all tracks. Participants may either evaluate their systems locally on the BAREC Dev and Test sets,[2] or submit their predictions on the BAREC Test set through the corresponding CodaBench competition for each track. Since the BAREC Test set is publicly available, anyone is welcome to participate in this phase.

In the **testing phase**, we release the Official Blind Test set exclusively to registered participants. Registered teams are required to submit system description papers.

Teams are permitted to participate in all tracks; however, their submissions must adhere to the resource constraints defined for each track. Participation in the constrained track is limited to the use of the SAMER corpus and/or lexicon, while participation in the open track requires the use of external resources.

5 Participants and Systems

We received 22 team registrations from 12 countries, of which 17 submitted system description papers. Table 6 lists the participating teams along with their affiliations and the tracks they joined. In total, we received 70 submissions during the development phase and 667 submissions during the testing phase. A detailed breakdown of submissions across tracks is provided in Table 8.

5.1 Baselines

We employed three baseline models to compare against the participating systems. All baselines are Arabic-specific BERT-base models fine-tuned on the BAREC Corpus, selected from the suite of models trained by Elmadani et al. (2025). These baselines vary along the following dimensions:

- Pretrained model: AraBERTv02 vs. AraBERTv2 (Antoun et al., 2020)
- Input variant: preprocessing of the BAREC Corpus - **Word** (simple sentence tokenization with diacritics and kashida removal) vs. **D3Tok** (tokenization of words into their base and clitic forms)[3]
- Loss function: Cross-entropy loss (CE) vs. Regression using Mean Squared Error (Reg)

Guided by these design choices, we selected the following three baselines:

[2] `https://github.com/CAMeL-Lab/barec-shared-task-2025`

[3] Preprocessing with CAMeL Tools (Obeid et al., 2020).

Team	Affiliation	Sent			Doc		
		S	C	O	S	C	O
!MSA (Basem et al., 2025)	MSA University, Egypt	✓	✓	✓	✓	✓	✓
AMAR (Saeed et al., 2025)	NYU Abu Dhabi, UAE	✓	✓		✓		
ANLPers (Sibaee et al., 2025)	Prince Sultan University, KSA	✓					
GNNinjas (Elchafei et al., 2025)	Ulm University, Germany		✓	✓		✓	
LIS (NAIT DJOUDI et al., 2025)	Aix Marseille Université, France	✓					
MARSAD (Ibrahim et al., 2025)	Northwestern University, Qatar	✓			✓		
MorphoArabia (Emad Eldin, 2025)	Cairo University, Egypt	✓	✓	✓	✓	✓	✓
mucAI (Abdou, 2025)	TUM, Germany	✓			✓		
Noor (Rabih, 2025)	MBZUAI, UAE	✓					
PalNLP (Ayesh, 2025)	Cardiff University, UK	✓					
Phantoms (Alhassan et al., 2025)	CMU Africa, Rwanda	✓					
Pixel (Sapirstein, 2025)	Reichman University, Israel	✓		✓			
Qais (Ahmed, 2025)	IMSIU, KSA	✓		✓			
SATLab (Bestgen, 2025)	UCLouvain, Belgique	✓			✓		
STBW (Trigui, 2025)	(Independent), UAE	✓			✓		
Syntaxa (Bahloul, 2025)	TUM, Germany	✓					
ZAI (Nazzal, 2025)	Zayed University, UAE	✓					

Table 6: List of participating teams, along with their affiliations and the tracks they participated in.

Team	Score	Features			Techniques						
		N-gram	Embeds	Morph	ML	PLM	LLM	Ord Loss	Ensemble	Label B.	GNN
!MSA	**87.5**			✓		✓	✓	✓	✓	✓	
AMAR	86.4			✓		✓	✓		✓		
mucAI	85.7			✓		✓			✓	✓	
STBW	85.6			✓		✓		✓			
ZAI	85.5			✓		✓		✓			
Baselines	84.6			✓		✓		✓			
Syntaxa	84.3		✓	✓		✓					✓
MorphoArabia	84.2			✓		✓		✓			
MARSAD	84.1			✓		✓		✓			
Noor	83.1			✓		✓					
Qais	83.0			✓		✓	✓			✓	
Phantom	82.7		✓	✓		✓					
LIS	82.4					✓					
SATLab	82.3	✓			✓						
PalNLP	81.1			✓		✓		✓	✓	✓	
GNNinjas	78.5		✓	✓		✓					✓
ANLPers	73.0		✓			✓		✓			
Pixel	68.4			✓		✓					

Table 7: Summary of features and techniques employed by participating teams with their best sentence-level QWK scores. *Embeds* refers to embeddings; *Morph* to morphological segmentation or features; *ML* to non-neural machine learning methods (e.g., SVMs); *PLM* to pre-trained language models; *LLM* to large language models used for prediction or data augmentation; *Ord Loss* to loss functions that account for the ordinal nature of labels (e.g., ordinal log loss, regression); *Label B.* to strategies addressing label imbalance; and *GNN* to graph neural networks.

Task	Development			Testing		
	S	C	O	S	C	O
Sent	63	1	2	221	78	98
Doc	2	1	1	110	83	77
		70			667	

Table 8: Number of valid submissions during the Development and testing phases across tracks.

- **AraBERTv02+Word+CE (Baseline I)**: serves as a standard baseline, combining the widely used AraBERTv02 with conventional word-level preprocessing and the standard cross-entropy loss.
- **AraBERTv2+D3Tok+CE (Baseline II)**: included to assess the effect of linguistically motivated tokenization (D3Tok) on classification performance.
- **AraBERTv2+D3Tok+Reg (Baseline III)**: motivated by the ordinal nature of readability levels, this baseline explores regression that accounts for the distance between predicted and true labels.

5.2 Summary of Submitted Systems

A summary of approaches employed by various teams is provided in Table 7. Most teams built on pre-trained language models (PLMs), often enhanced with morphological features, while a few also incorporated ensembling, label balancing, or ordinal-aware loss functions. Teams such as !MSA, AMAR, and Qais further leveraged large language models (LLMs), and graph neural networks (GNNs) (Zhou et al., 2021) were explored by GNNinjas and Syntaxa, while team Pixel explored vision language models (Dosovitskiy et al., 2021; Rust et al., 2023). Traditional non-neural methods were rare. Overall, the strongest approaches combined PLMs with linguistic features and ensembling. Next we present the system description of the best performing team.

5.3 !MSA: Best Performing Team

The pipeline of the winning team, !MSA (Basem et al., 2025), begins by preprocessing the data with D3Tok tokenization (Obeid et al., 2020). They further augment the data differently for each track: upsampling for the strict track, SAMER corpus-based augmentation for the constrained track, and paraphrasing 12k entries from the BAREC corpus using the Gemini API for the open track.[4] Beyond preprocessing, the pipeline is consistent across all tracks. They employ an ensemble of models — AraBERTv2 (Antoun et al., 2020), AraElectra (Antoun et al., 2021), MARBERT (Abdul-Mageed et al., 2021), and CamelBERT (Inoue et al., 2021) — trained with different loss functions including Cross-Entropy, Ordinal Log Loss (Castagnos et al., 2022), Regression (Mean Squared Error), and Conditional Ordinal Regression (Cao et al., 2020). The ensemble combines model outputs via a weighted average, where weights are determined based on each model's confidence scores.

The following section provides a general discussion of the results and analyzes how different approaches impacted performance.

6 Results

Tables 9 and 10 show the results in QWK for all tracks in the sentence-level and document-level tasks, respectively. The baselines, highlighted in gray, were trained only on the BAREC corpus, and their scores are reported identically across tracks to facilitate comparison with other teams. Overall, five teams outperformed our strongest baseline. In most cases, however, the additional resources available in the constrained and open tracks did not yield improvements over the strict track. Team **!MSA** achieved the highest QWK scores across all tasks and tracks. In this section, we provide a broad analysis of the overall results. We also report on the other metrics in Appendix A.

6.1 General Discussion

Table 7 summarizes the participating teams, their best scores, and the features and techniques they employed. The results show a clear dominance of pre-trained language models (PLMs), which were adopted by nearly all teams. The top performers — !MSA (87.5), AMAR (86.4), and mucAI (85.7) — achieved their results by training on morphologically segmented text and combining PLMs with ordinal-aware loss functions and strategies for addressing label imbalance. Ensembling further boosted performance, particularly for the leading teams, by allowing them to leverage multiple models. Morphological features were widely used across systems, underscoring the importance of morphology-aware approaches in Arabic readabil-

[4] https://ai.google.dev/

ity assessment. Team **!MSA** led the leaderboard by integrating all of these components. In contrast, traditional machine learning methods and n-gram features were rarely employed and, when used, did not yield competitive results. More modern approaches such as graph neural networks (GNNs) and vision language models (e.g. team Pixel) also failed to provide significant gains. Large language models (LLMs) were explored by three teams for prediction and data augmentation, but in both cases did not outperform PLM-based systems. Overall, the findings highlight that success in this task depended primarily on combining morphological segmentation with PLMs, ensembling, and ordinal-sensitive modeling.

Team	Strict	Constrained	Open
!MSA	**87.5**	**86.6**	**86.4**
AMAR	86.4	86.4	
mucAI	85.7		
STBW	85.6		
ZAI*	85.5		
Baseline III	84.6	84.6	84.6
Syntaxa	84.3		
MorphoArabia	84.2	82.9	83.9
MARSAD	84.1		
Noor	83.1		
Phantom	82.7		
Qais	82.5		83.0
LIS	82.4		
SATLab	82.3		
Baseline II	81.5	81.5	81.5
PalNLP	81.1		
Baseline I	80.5	80.5	80.5
ANLPers*	73.0		
Pixel	66.2		68.4
GNNinjas		78.5	77.6

Table 9: Performance of participating teams across all tracks in the **sentence-level** task. Scores are reported as QWK (%) and sorted based on the performance on the strict track. * denotes systems that used the dev set for training, making their scores not directly comparable to others.

Team	Strict	Constrained	Open
!MSA	**87.4**	**84.3**	**82.2**
MorphoArabia	79.9	75.5	79.2
MARSAD	79.0		
SATLab	77.6		
mucAI	73.3		
Baseline III	72.6	72.6	72.6
STBW	72.5		
AMAR	69.6		
Baseline II	62.0	62.0	62.0
Baseline I	57.7	57.7	57.7
GNNinjas		76.9	

Table 10: Performance of participating teams across all tracks in the **document-level** task. Scores are reported as QWK (%) and sorted based on the performance on the strict track.

7 Conclusion

In this paper, we presented the framework and results of the BAREC 2025 Shared Task on Fine-Grained Arabic Readability Assessment—the first shared task dedicated to this problem. The task featured two subtasks (sentence-level and document-level) and three tracks (strict, constrained, and open). A new blind test set was created for the evaluation, consisting of 3,420 sentences and 100 documents. In total, 22 teams from 12 countries registered, and 17 submitted system description papers. The strong participation highlights the interest in Arabic readability assessment. Looking ahead, we plan to expand available resources and organize future shared tasks to further advance research in this area.

Limitations

This work has a few limitations worth noting. First, document-level readability was derived from sentence-level readability under the assumption that the hardest sentence determines the overall document level. While simple, this approach often pushes documents toward higher readability levels, since a single difficult sentence can raise the document's level. Second, we adopted Quadratic Weighted Kappa (QWK) as the primary evaluation metric. However, the choice of the most suitable metric for this task remains an open question.

Acknowledgments

The BAREC project is supported by the Abu Dhabi Arabic Language Centre (ALC) / Department of Culture and Tourism, UAE.

References

Ahmed Abdou. 2025. mucAI at BAREC Shared Task 2025: Towards Uncertainty Aware Arabic Readability Assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

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.

Samar Ahmed. 2025. Qais at BAREC Shared Task 2025: A Fine-Grained Approach for Arabic Readability Classification Using a pre-trained model. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

M Al-Dawsari. 2004. The assessment of readability books content (boys-girls) of the first grade of intermediate school according to readability standards. *Sultan Qaboos University, Muscat*.

Hend S Al-Khalifa and Amani A Al-Ajlan. 2010. Automatic readability measurements of the Arabic text: An exploratory study. *Arabian Journal for Science and Engineering*, 35(2 C):103–124.

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Ahmed Alhassan, Asim Mohamed, and Moayad Elamin. 2025. Phantoms at BAREC Shared Task 2025: Enhancing Arabic Readability Prediction with Hybrid BERT and Linguistic Features. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Richard L Allington, Kimberly McCuiston, and Monica Billen. 2015. What research says about text complexity and learning to read. *The Reading Teacher*, 68(7):491–501.

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mutaz Ayesh. 2025. PalNLP at BAREC Shared Task 2025: Predicting Arabic Readability Using Ordinal Regression and K-Fold Ensemble Learning. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Ahmed Bahloul. 2025. Syntaxa at BAREC Shared Task 2025: BERTnParse - Fusion of BERT and Dependency Graphs for Readability Prediction. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Amelia T. Barber and Susan L. Klauda. 2020. How reading motivation and engagement enable reading achievement: Policy implications. *Policy Insights from the Behavioral and Brain Sciences*, 7(1):27–34.

Mohamed Basem, Mohammed Younes, Seif Ahmed, and Abdelrahman Moustafa. 2025. !MSA at BAREC Shared Task 2025: Ensembling Arabic Transformers for Readability Assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Eman Zahran, Hamdo Elhuseyin, Tamer Elsayed, Walid Massoud, and Houda Bouamor. 2024. Qaes: First publicly-available trait-specific annotations for automated scoring of arabic essays. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 337–351.

Yves Bestgen. 2025. SATLab at BAREC Shared Task 2025: Optimizing a Language-Independent System for Fine-Grained Readability Assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331.

François Castagnos, Martin Mihelich, and Charles Dognin. 2022. A simple log-based loss function for ordinal text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4604–4609, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Kevyn Collins-Thompson and James P. Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proceedings of the Human Language Technology Conference of the North American*

Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, pages 193–200, Boston, Massachusetts, USA. Association for Computational Linguistics.

C. o. E. Council of Europe. 2001. Common european framework of reference for languages: learning, teaching, assessment.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17, Seattle, WA, USA → Online. Association for Computational Linguistics.

Afrizal Doewes, Nughthoh Arfawi Kurdhi, and Akrati Saxena. 2023. Evaluating quadratic weighted kappa as the standard performance metric for automated essay scoring. In *Proceedings of the 16th International Conference on Educational Data Mining*, pages 103–113, Bengaluru, India. International Educational Data Mining Society.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

William H DuBay. 2004. The principles of readability. *Online Submission*.

Mahmoud El-Haj and Paul Rayson. 2016. OSMAN — a novel Arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255, Portorož, Slovenia. European Language Resources Association (ELRA).

Mo El-Haj, Sultan Almujaiwel, Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2024. DARES: Dataset for Arabic readability estimation of school materials. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 103–113, Torino, Italia. ELRA and ICCL.

Passant Elchafei, Mayar Osama, Mohamed Rageh, and Mervat Abu-Elkheir. 2025. GNNinjas at BAREC Shared Task 2025: Lexicon-Enriched Graph Modeling for Arabic Document Readability Prediction. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Fatimah Mohamed Emad Eldin. 2025. MorphoArabia at BAREC Shared Task 2025: A Hybrid Architecture with Morphological Analysis for Arabic Readability Assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Coling 2010: Posters*, pages 276–284, Beijing, China. Coling 2010 Organizing Committee.

Jonathan Forsyth. 2014. Automatic readability prediction for modern standard Arabic. In *Proceedings of the Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*.

Irene C Fountas and Gay Su Pinnell. 2006. *Leveled books (k-8): Matching texts to readers for effective teaching*. Heinemann Educational Books.

Nizar Habash and David Palfreyman. 2022. ZAEBUC: An annotated Arabic-English bilingual writer corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 79–88, Marseille, France. European Language Resources Association.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Shimaa Ibrahim, Md. Rafiul Biswas, Mabrouka Bessghaier, and Wajdi Zaghouani. 2025. MarsadLab at BAREC Shared Task 2025: Strict-Track Readability Prediction with Specialized AraBERT Models on BAREC. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. 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.

Adam Kilgariff, Frieda Charalabopoulou, Maria Gavrilidou, Janne Bondi Johannessen, Saussan Khalil, Sofie Johansson Kokkinakis, Robert Lew, Serge Sharoff, R. Vadlapudi, and Elena Volodina. 2014. Corpus-based vocabulary lists for language learners for nine languages. *Language resources and evaluation*, 48(1):121–163.

G.R. Klare. 1963. *The Measurement of Readability*. Iowa State University Press.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Farah Nadeem and Mari Ostendorf. 2018. Estimating linguistic complexity for science texts. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 45–55, New Orleans, Louisiana. Association for Computational Linguistics.

Anya Amel NAIT DJOUDI, Patrice Bellot, and Adrian-Gabriel Chifu. 2025. LIS at BAREC Shared Task 2025: Multi-Scale Curriculum Learning for Arabic Sentence-Level Readability Assessment Using Pretrained Language Models. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Tarek Naous, Michael J Ryan, Anton Lavrouk, Mohit Chandra, and Wei Xu. 2024. ReadMe++: Benchmarking multilingual language models for multi-domain readability assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12230–12266, Miami, Florida, USA. Association for Computational Linguistics.

Naoual Nassiri, Violetta Cavalli-Sforza, and Abdelhak Lakhouaja. 2023. Approaches, methods, and resources for assessing the readability of arabic texts. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Ahmad M. Nazzal. 2025. ZAI at BAREC Shared Task 2025: AraBERT CORAL for Fine Grained Arabic Readability. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. 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.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.

Nour Rabih. 2025. Noor at BAREC Shared Task 2025: A Hybrid Transformer-Feature Architecture for Sentence-level Readability Assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. *Preprint*, arXiv:2207.06991.

Hind Saddiki, Nizar Habash, Violetta Cavalli-Sforza, and Muhamed Al Khalil. 2018. Feature optimization for predicting readability of arabic l1 and l2. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 20–29.

Mostafa Saeed, Rana Waly, and Abdelaziz Ashraf Hussein. 2025. AMAR at BAREC Shared Task 2025: Arabic Meta-learner for Assessing Readability. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Ben Sapirstein. 2025. Pixels at BAREC Shared Task 2025: Visual Arabic Readability Assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Serry Sibaee, Omer Nacar, Yasser Alhabashi, Adel Ammar, Yasser Al-Habashi, and Wadii Boulila. 2025. ANLPers at BAREC Shared Task 2025: Readability of Embeddings Training Neural Readability Classifiers on the BAREC Corpus. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Rasha Soliman and Laila Familiar. 2024. Creating a cefr arabic vocabulary profile: A frequency-based multi-dialectal approach. *Critical Multilingualism Studies*, 11(1):266–286.

Hanada Taha-Thomure. 2017. *Arabic Language Text Leveling* (معايير هنادا طه لتصنيف مستويات النصوص العربية). Educational Book House (دار الكتاب التربوي للنشر والتوزيع).

Saoussan Trigui. 2025. STBW at BAREC Shared Task 2025: AraBERT-v2 with MSE-SoftQWK Loss for Sentence-Level Arabic Readability. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Sowmya Vajjala. 2022. Trends, limitations and open challenges in automatic readability assessment research. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5366–5377, Marseille, France. European Language Resources Association.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 163–173, Montréal, Canada. Association for Computational Linguistics.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Zhen Xu, Sergio Escalera, Adrien Pavão, Magali Richard, Wei-Wei Tu, Quanming Yao, Huan Zhao, and Isabelle Guyon. 2022. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns*, 3(7):100543.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2021. Graph neural networks: A review of methods and applications. *Preprint*, arXiv:1812.08434.

A Additional Results

Team	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**	**Acc^{7}**	**Acc^{5}**	**Acc^{3}**
!MSA	**87.5**	43.5	**76.7**	**1.0**	64.1	69.6	76.2
AMAR	86.4	39.7	73.2	1.1	60.8	67.8	76.1
mucAI*	85.7	50.9	75.6	**1.0**	65.2	69.8	76.1
STBW	85.6	33.3	73.6	1.2	57.3	66.5	74.7
ZAI	85.5	48.8	73.3	**1.0**	64.4	69.3	75.8
Syntaxa	84.3	51.0	72.0	**1.0**	64.4	68.7	75.4
MorphoArabia	84.2	43.5	74.0	1.1	63.0	68.3	75.3
MARSAD	84.1	52.0	74.0	**1.0**	65.9	70.6	76.1
Noor	83.1	56.1	72.5	**1.0**	67.0	70.5	75.8
Phantom	82.7	**57.6**	72.3	**1.0**	67.4	71.3	**77.2**
Qais	82.5	54.8	71.8	1.1	65.1	69.5	75.3
LIS	82.4	57.5	72.4	**1.0**	**67.8**	**71.5**	76.4
SATLab	82.3	25.8	63.1	1.4	47.0	59.0	69.8
PalNLP*	81.1	33.1	69.8	1.3	57.2	63.6	72.5
ANLPers	73.0	44.7	61.4	1.4	56.3	62.0	70.0
Pixel	66.2	38.1	53.6	1.8	48.6	54.2	65.9

Table 11: Performance of participating teams in the **strict** track in the **sentence-level** task. Results are sorted based on the QWK score. * denotes systems that used the dev set for training, making their scores not directly comparable to others.

Team	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**	**Acc^{7}**	**Acc^{5}**	**Acc^{3}**
!MSA	**86.6**	44.9	**75.4**	**1.0**	**63.0**	**68.7**	75.6
AMAR	86.4	39.9	73.0	1.1	61.0	68.1	**76.3**
MorphoArabia	82.9	30.9	70.6	1.3	53.9	62.9	72.1
GNNinjas	78.5	**50.0**	67.2	1.4	61.2	66.1	74.9

Table 12: Performance of participating teams in the **constrained** track in the **sentence-level** task. Results are sorted based on the QWK score.

Team	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**	**Acc^{7}**	**Acc^{5}**	**Acc^{3}**
!MSA	**86.4**	41.3	**75.1**	**1.0**	61.7	67.3	74.5
MorphoArabia	83.9	48.8	71.3	1.1	62.5	67.6	74.3
Qais	83.0	**54.2**	71.8	1.1	**66.0**	**70.0**	**75.8**
GNNinjas	77.6	48.7	66.5	1.3	60.7	65.2	74.5
Pixel	68.4	41.5	56.8	1.6	50.9	56.8	65.1

Table 13: Performance of participating teams in the **open** track in the **sentence-level** task. Results are sorted based on the QWK score.

Team	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**	**Acc^{7}**	**Acc^{5}**	**Acc^{3}**
!MSA	**87.4**	**52**	**94**	**0.6**	**81**	**81**	**93**
MorphoArabia	79.9	42	90	0.7	71	71	92
MARSAD	79.0	36	84	0.8	59	60	85
SATLab	77.6	39	88	0.8	70	71	87
mucAI	73.3	36	86	0.8	65	66	89
STBW	72.5	35	85	0.8	67	67	90
AMAR	69.6	34	79	0.9	70	70	89

Table 14: Performance of participating teams in the **strict** track in the **document-level** task. Results are sorted based on the QWK score.

Team	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**	**Acc^{7}**	**Acc^{5}**	**Acc^{3}**
!MSA	**84.3**	**48**	**91**	**0.6**	**77**	**77**	**94**
GNNinjas	76.9	42	83	0.8	60	61	90
MorphoArabia	75.5	34	83	0.9	64	65	85

Table 15: Performance of participating teams in the **constrained** track in the **document-level** task. Results are sorted based on the QWK score.

Team	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**	**Acc^{7}**	**Acc^{5}**	**Acc^{3}**
!MSA	**82.2**	**50**	**86**	**0.6**	**70**	**70**	89
MorphoArabia	79.2	37	**86**	0.8	65	65	**92**

Table 16: Performance of participating teams in the **open** track in the **document-level** task. Results are sorted based on the QWK score.

Syntaxa at BAREC Shared Task 2025: BERTnParse - Fusion of BERT and Dependency Graphs for Readability Prediction

Ahmed Bahloul
School of Computation, Information and Technology, Technical University of Munich
Munich, Germany
ahmed.bahloul@tum.de

Abstract

We describe our system submission to **Task 1 (Sentence-level Readability Assessment)** of the BAREC Shared Task 2025 (Elmadani et al., 2025a), in the **strict track**. Task 1 requires predicting the readability level of an Arabic sentence on a scale from 1 (easiest) to 19 (hardest), reflecting reading difficulty. Our approach integrates contextual and syntactic information by combining pretrained BERT embeddings (Devlin et al., 2019) with a Graph Neural Network (GNN) (Zhou et al., 2021) over dependency parse trees (Kipf and Welling, 2017). Our hypothesis is that readability is influenced not only by word choice but also by syntactic complexity—especially in morphologically rich languages like Arabic (Habash, 2010). To capture both aspects, we represent each sentence as a dependency graph with BERT token embeddings as node features, and use a GNN to model the syntactic structure. Experimental results show that our syntax-aware model improves over a strong BERT baseline, highlighting the value of structural linguistic information for fine-grained readability classification.[1]

1 Introduction

Readability assessment aims to estimate the difficulty of a text for a given audience. For Arabic, this task is particularly challenging due to the language's rich morphology, flexible word order, and cliticization. Sentence-level readability prediction demands models that capture subtle syntactic and semantic cues. While transformer-based models like AraBERTv2 (Antoun et al., 2020) encode deep lexical features, they often underutilize syntactic structure—an important aspect of textual complexity.

We propose a hybrid architecture that integrates *syntactic dependency graphs* with *contextual embeddings* from AraBERTv2 for Arabic sentence-level readability prediction. Dependency trees are parsed into graphs and processed with a Graph Neural Network (GNN), while token embeddings from AraBERTv2 are used to represent semantic content. The resulting model jointly reasons over both syntactic and contextual signals.

While GNNs have been combined with transformers in tasks like QA (Yasunaga et al., 2021), document classification (Zhang et al., 2020), and semantic role labeling (Marcheggiani and Titov, 2017), such architectures have not been applied to **readability assessment**. For Arabic, prior work relies on feature-based or PLM-only methods (Liberato et al., 2024) or on word-level readability tools (Hazim et al., 2022; Al Khalil et al., 2020), rather than sentence-level prediction. Recent multilingual efforts such as ReadMe++ (Naous et al., 2024) evaluate both supervised and prompting-based methods, as well as unsupervised approaches, but all rely solely on pretrained language models without incorporating syntactic structure. This leaves the impact of syntax underexplored—especially in morphologically rich languages. The closest related work is by Ivanov (Ivanov, 2022), who compares syntax-based GNNs using fastText embeddings and BERT-based models for sentence complexity in Russian, but does not integrate syntactic and contextual representations into a unified model.

In addition to our architecture, we propose a novel alignment strategy that merges AraBERT subword embeddings and dependency parse nodes into **word-level units**. This is crucial for Arabic, where clitics and morphology lead to tokenization mismatches. Prior work often sidesteps this mismatch by propagating labels across subwords, but we instead ensure structural and semantic alignment through node merging and embedding pooling, enabling effective message passing in the graph.

Contributions: (1) We propose a syntax-aware model that fuses GNN-based syntactic representa-

[1] Code available at: https://github.com/ahmedehabb/BERTnParse

Proceedings of The Third Arabic Natural Language Processing Conference, pages 253–260
November 8-9, 2025 ©2025 Association for Computational Linguistics

Level	Arabic	Transliteration (HSB)	English
6	هنا يلتقي ماجد كل أسبوع بأصدقائه	hnA yltqy mAjd kl Âsbwς bÂSdqAŷh	Here Majid meets his friends every week.
11	ما هو المقصود دول عدم الانحياز؟	mA hw AlmqSwd dwl ςdm AlAnHyAz?	What is meant by the Non-Aligned Movement?

Table 1: Example BAREC sentences with readability levels, CAMeL Tools HSB transliterations, and English glosses.

tions with AraBERTv2 for Arabic sentence-level readability. (2) We introduce a word-level alignment method addressing tokenization mismatches between BERT and dependency parses. (3) We improve over a strong BERT baseline on the BAREC corpus (Elmadani et al., 2025b), especially for complex sentences.

2 Data

We evaluate our approach on the **Balanced Arabic Readability Evaluation Corpus (BAREC)** (Elmadani et al., 2025b), released as part of the BAREC Shared Task 2025. The dataset comprises Arabic sentences labeled with **19 readability levels** (1 = easiest, 19 = hardest), covering diverse topics and genres.

The dataset is split as follows:

- **Train set**: 54,845 sentences
- **Dev set**: 7,310 sentences
- **Test set**: 7,286 sentences
- **Blind Test set**: 3,420 sentences

Each sentence is annotated with a readability level following detailed linguistic and pedagogical guidelines (Habash et al., 2025). To illustrate the dataset, we provide examples with their readability levels, CAMeL Tools HSB transliterations (Habash et al., 2007), and English glosses.[2] Table 1 shows two representative examples.

Additional Arabic readability resources, such as the SAMER corpus (Alhafni et al., 2024), may be useful for future research. However, in line with the **strict track** guidelines of the BAREC Shared Task—where models must be trained exclusively on the BAREC training set—we restrict our experiments to the BAREC dataset only.

3 Methodology

Our approach combines contextual embeddings from AraBERTv2 (Antoun et al., 2020) with a Graph Neural Network (GNN) applied to the syntactic dependency graph of each input sentence. This design enables the model to jointly capture lexical semantics and syntactic structure, addressing key challenges in fine-grained Arabic readability prediction.

3.1 Input Representation

Given an input Arabic sentence $S = (w_1, w_2, \dots, w_n)$, we first obtain token-level contextual embeddings $\mathbf{h}_i^{\text{BERT}} \in \mathbb{R}^d$ from AraBERTv2, where d is the embedding dimension. AraBERTv2 parameters are **fine-tuned** during training to adapt to the readability task.

Simultaneously, we parse S using Camel-Parser2.0 (Elshabrawy et al., 2023) to obtain a dependency graph $G = (V, E)$, where $V = \{w_i\}$ are nodes corresponding to tokens, and edges $E \subseteq V \times V$ represent syntactic relations. Each node $w_i \in V$ is also associated with a part-of-speech (POS) tag, and each edge $e = (w_i \to w_j) \in E$ is labeled with a dependency type (e.g., `OBJ`, `SUBJ`).

3.2 Token Alignment and Word-Level Processing

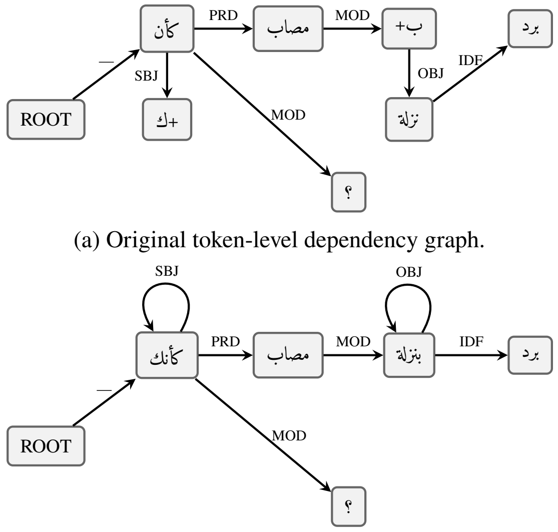
A key design decision in our system is to operate at the *word level* rather than on subword units. While AraBERTv2 employs WordPiece tokenization (Wu et al., 2016), which splits words into subword segments, the dependency parser outputs nodes that often correspond to grammatical morphemes or clitics. For example, the Arabic word سأصيدهما (transliteration: sÂSydhmA, translation: “I will catch them”) is segmented into the future tense particle +س (s+, PART), the verb stem أصيد (Aṣyd, VRB), and the object pronoun suffix هما+ (+hmA, NOM). This linguistically motivated segmentation differs from the subword units generated by BERT,

[2] Transliteration via the CAMeL Tools CLI: `camel_transliterate -s ar2hsb < file`

which are learned based on frequency statistics.

To resolve this mismatch, we average the AraBERTv2 subword embeddings into a single word-level vector, following the general idea of leveraging subword information for richer word representations (Bojanowski et al., 2017). On the parsing side, we merge subword nodes (e.g., clitics) and their associated edges into unified word-level graph nodes. This ensures that each word is consistently represented by both a single graph node and a single embedding. Figure 1 illustrates this process by comparing the original token-level dependency graph with the merged word-level version. A detailed description of the merging procedure, along with transliteration and English glosses for the example sentence, is provided in Section 3.3.

This alignment is critical for ensuring consistent and interpretable graph structures. It eliminates discrepancies between the number of embedding vectors and graph nodes, enabling meaningful message passing and feature aggregation in the GNN. To our knowledge, this approach to harmonizing tokenization granularity in Arabic is novel and effectively addresses challenges posed by the language's rich morphology and syntactic structure.



(a) Original token-level dependency graph.

(b) Merged word-level dependency graph.

Figure 1: Comparison between token-level and merged word-level dependency graphs for the Arabic phrase كأنك مصاب بنزلة برد؟

3.3 Subword-to-Word Merging for Dependency Graphs

We parsed the Arabic sentence كأنك مصاب بنزلة برد؟ using CAMeLParser2.0, which outputs token-level dependencies, including affixes and clitics as separate units.

Sentence example with transliteration and translation:

Original: كأنك مصاب بنزلة برد؟
Transliteration: kÂnk mSAb bnzlh brd?
Translation: As if you have a cold?

Token list:

1. كأن — PRT (base)
2. +ك — PRON (enclitic)
3. مصاب — ADJ
4. +ب — ADP (prefix)
5. نزلة — NOUN
6. برد — NOUN
7. ? — PUNCT

Original edges (token-level): Token ID 0 refers to the artificial ROOT node.

- (1 → 0) —
- (2 → 1) SBJ
- (3 → 1) PRD
- (4 → 3) MOD
- (5 → 4) OBJ
- (6 → 5) IDF
- (7 → 1) MOD

Merging subword tokens: We merged:

- Tokens [1,2] → كأنك
- Tokens [4,5] → بنزلة

All incoming and outgoing edges of the merged tokens are also combined, so that the resulting word node preserves the original dependency relations for consistent graph construction.

Graph Construction We construct the input graph for the GNN as follows:

- **Nodes and node features**: Each node corresponds to a token w_i and is initialized with the corresponding BERT embedding $\mathbf{h}_i^{\text{BERT}}$. POS tags are encoded as learnable embeddings and concatenated to the token representations. The special `[CLS]` token is used to represent the syntactic root of the sentence and serves as the head node in the graph.

- **Edges and edge features**: Directed edges are constructed based on the dependency parse. Each edge is labeled with a dependency relation, which is encoded as a learnable embedding $\mathbf{e}_{\text{rel}}$ and incorporated via edge-aware message passing.

3.4 GNN Architecture and Training

We employ a multi-layer Graph Neural Network based on `TransformerConv` (Shi et al., 2021) layers to propagate syntactic information. Each TransformerConv layer uses multi-head self-attention (4 heads) over nodes, with attention scores modulated by edge attributes.

At layer l, the hidden state of node i is updated by attending over its neighbors $\mathcal{N}(i)$, conditioning on both node features and edge embeddings. Aggregated edge embeddings for incoming and outgoing edges are computed separately and fused into node representations through a linear projection. Formally, the node representations evolve as:

$$\mathbf{h}_i^{(l+1)} = \texttt{TransformerConv}\big(\mathbf{h}_i^{(l)}, \{\mathbf{h}_j^{(l)} : j \in \mathcal{N}(i)\}, \mathbf{e}_{rel}\big)$$

where $\mathbf{e}_{rel}$ are learned edge embeddings processed by an MLP.

After stacking TransformerConv layers with dropout and layer normalization, node features are aggregated via attentional pooling to produce a graph-level embedding $\mathbf{h}_S$.

This embedding is then fed into two parallel fully connected layers, generating logits for two complementary objectives:

$$\mathbf{z}_{\text{CORAL}} = \mathbf{W}_{\text{c}}\mathbf{h}_S + \mathbf{b}_{\text{c}}, \quad \mathbf{z}_{\text{QWK}} = \mathbf{W}_{\text{q}}\mathbf{h}_S + \mathbf{b}_{\text{q}}.$$

The first layer outputs 18 logits corresponding to ordinal thresholds for the CORAL loss (Cao et al., 2020), while the second produces 19 logits for direct classification used by the Quadratic Weighted Kappa (QWK) loss (de La Torre et al., 2018).

To balance ordinal accuracy and agreement quality, we optimize a combined loss:

$$\mathcal{L} = 0.5 \cdot \mathcal{L}_{\text{CORAL}} + 0.5 \cdot \mathcal{L}_{\text{QWK}},$$

where $\mathcal{L}_{\text{QWK}}$ penalizes larger prediction errors more heavily, enhancing robustness to class imbalance and ordinal inconsistencies.

Figure 2 illustrates the overall model pipeline, highlighting the integration of AraBERTv2 and syntactic parsing through a GNN layer for joint representation learning.

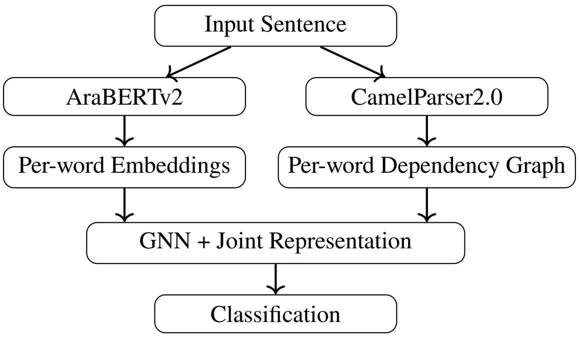


Figure 2: Model architecture integrating lexical and syntactic information for readability prediction.

4 Experimental Setup

4.1 Modeling and Preprocessing

We preprocess each sentence using WordPiece tokenization from AraBERTv2 and dependency parsing via `CamelParser2.0`, which outputs POS tags, syntactic relations, and token-level dependency structures. Following the word-to-subtoken alignment procedure detailed in 3.2, we average subtoken embeddings to form word-level representations. Correspondingly, subtoken-based nodes in the dependency graph are merged into single word-level nodes. The special `[CLS]` token is used to represent the syntactic root of the graph and serves as the anchor node for the sentence-level structure.

Our model integrates these word-level embeddings and dependency graphs through a 4-layer `TransformerConv`-based Graph Neural Network (GNN) with a hidden size of 512. We incorporate learnable embeddings for POS tags and dependency relations. Training is performed with the Adam optimizer using a learning rate of 1×10^{-4}, batch size of 64, dropout rate of 0.2, and early stopping based on validation loss.

Test Set	Model	QWK	Accuracy	Acc ±1	Dist	Acc 7	Acc 5	Acc 3
Internal	Baseline	80.2	**55.9%**	70.0%	1.1	**65.1%**	**69.4%**	**75.2%**
	Our Model	**83.7**	50.5%	**71.9%**	**1.0**	63.1%	68.0%	74.2%
Official Blind	Baseline	81.5	**58.1%**	**72.0%**	**1.0**	**67.7%**	**71.4%**	**76.5%**
	Our Model	**84.3**	51.0%	**72.0 %**	**1.0**	64.4%	68.7%	75.4%

Table 2: Performance on internal and official blind test sets for sentence-level readability prediction.

4.2 Evaluation Metrics

We treat readability assessment as an ordinal classification task. Our primary metric is **Quadratic Weighted Kappa (QWK)**, which penalizes larger prediction errors quadratically. We also report Exact Match Accuracy (Acc19) on the 19-level scale, along with adjacent accuracy (±1), coarser-grained accuracies (Acc7, Acc5, Acc3), and average prediction distance measured by mean absolute error.

5 Results

5.1 Comparison of Model Variants

We evaluate two model variants to assess the contribution of syntactic and structural information:

- **AraBERTv2 baseline**: Fine-tuned on the BAREC-Corpus-v1.0 Word input using cross-entropy loss (Elmadani et al., 2025b).
- **AraBERTv2 + GNN (ours)**: Our proposed approach integrates syntactic dependency parsing using a TransformerConv-based Graph Neural Network over word-level BERT embeddings. Each word node is enriched with POS and syntactic edge features, and the special `[CLS]` token anchors the graph as the syntactic root.

5.2 Performance on Internal and Official Test Sets

Table 2 shows our model achieves superior Quadratic Weighted Kappa (QWK) scores on both internal (83.7 vs. 80.2) and official blind test sets (84.3 vs. 81.5) compared to the baseline, indicating stronger ordinal agreement.

Our method also yields lower average prediction distances (1.0 vs. 1.1 internally) and competitive adjacent accuracy (±1), suggesting more calibrated and consistent predictions. While the baseline slightly outperforms in strict exact match accuracy, our model's improvements in ordinal metrics underscore the benefits of integrating syntactic structure.

6 Conclusion and Future Work

In this work, we explored the integration of contextual semantic features from AraBERTv2 with syntactic structure captured via dependency parsing graphs for the task of Arabic sentence-level readability assessment. Our model incorporates a TransformerConv-based GNN over a dependency graph constructed at the word level, resolving alignment inconsistencies between WordPiece tokenization and morphological segmentation. We demonstrated that augmenting AraBERTv2 with structural information significantly improves performance over a strong BERT-only baseline. Our findings highlight the value of syntactic context in modeling Arabic linguistic complexity and offer a promising direction for fine-grained readability prediction in morphologically rich languages.

For future work, we plan to investigate the use of multilingual pretrained models to leverage cross-lingual knowledge and improve generalization across different Arabic dialects and related languages. Additionally, exploring alternative architectures beyond encoder-only models, such as encoder-decoder or graph transformers, may further enhance the integration of syntactic and semantic information for readability prediction.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

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.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331.

Jordi de La Torre, Domenec Puig, and Aida Valls. 2018. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognition Letters*, 105:144–154.

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.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

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.

Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Nizar Habash, Abdelhadi Soudi, and Timothy Buckwalter. 2007. *On Arabic Transliteration*, pages 15–22. Springer Netherlands, Dordrecht.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Vladimir Vladimirovich Ivanov. 2022. Sentence-level complexity in russian: An evaluation of bert and graph neural networks. *Frontiers in Artificial Intelligence*, 5:1008411.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *Preprint*, arXiv:1609.02907.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Tarek Naous, Michael J Ryan, Anton Lavrouk, Mohit Chandra, and Wei Xu. 2024. ReadMe++: Benchmarking multilingual language models for multi-domain readability assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12230–12266, Miami, Florida, USA. 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.

Xintong Shi, Wenzhi Cao, and Sebastian Raschka. 2023. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2021. Masked label prediction: Unified message passing model for semi-supervised classification. *Preprint*, arXiv:2009.03509.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff

Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, and 12 others. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *Preprint*, arXiv:1609.08144.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 334–339, Online. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2021. Graph neural networks: A review of methods and applications. *Preprint*, arXiv:1812.08434.

A Appendix

A.1 Reproducibility Details

A.1.1 Data Preprocessing

- Sentences are cleaned and tokenized using CAMeL Tools.

- We additionally remove *tatweel* characters (ـ) from the text, as many were found to be broken or incorrectly inserted in the middle of words, which can negatively affect tokenization and model performance. For example, the word مدرّسة might appear as مدرّسـ ـة, which we normalize by removing the tatweel and any extra spaces to restore the correct word form.

- POS tag IDs and dependency relation IDs are mapped using predefined dictionaries (`pos2id`, `dep2id`), which we constructed from the train set to cover all observed tags and relations.

- The graph is stored using PyTorch Geometric's `Data` objects with fields: `x`, `edge_index`, `edge_attr`, `pos_tag_ids`, and custom fields like `sentence`.

A.1.2 Model Components

1. AraBERTv2 (Encoder)

- Pretrained weights loaded from `aubmindlab/bert-base-arabertv02`.

- WordPiece tokenization applied via HuggingFace tokenizer.

- Hidden size: 768.

- For each token, embeddings are obtained by mean-pooling over all subtokens aligned via `encoding.word_ids()`, using the mean of the last 4 hidden layers' outputs.

- The first 8 layers of AraBERTv2 are frozen during training, and only the last 4 layers are fine-tuned.

2. Graph Construction

- Sentences are tokenized using CAMeL Tools' (Obeid et al., 2020) morphological segmenter.

- Dependency parses are extracted via the CamelParser2.0.

- For each sentence:
  - Nodes represent surface-level word tokens (segmented, not subword).
  - Directed edges represent syntactic dependencies (head → dependent).
  - Each edge is labeled by the dependency relation (e.g., `SBJ`, `OBJ`).
  - Part-of-speech (POS) tags are extracted per token.
  - The token labeled as `ROOT` by the parser is treated as the syntactic head of the sentence and serves as the root of the dependency tree.

3. Graph Neural Network Architecture Details

- **Node input:** Concatenation of AraBERTv2 embedding and averaged POS tag embedding (32-dimensional).

- **Edge input:** Relation type embedding (hidden size / 2), passed through a feedforward projection.

- **Convolution:** 4-layer `TransformerConv` (with 4 heads), using edge features in attention.

- **Edge Aggregation:** Mean aggregation of outgoing and incoming edge features per node.
- **Normalization:** LayerNorm applied after each GNN layer.
- **Pooling:** `AttentionalAggregation` over graph-level node embeddings.
- **Classifier heads:** One linear layer with $C - 1$ units for CORAL ordinal regression, and a separate linear head with C units for optimizing the Quadratic Weighted Kappa (QWK) loss.

A.2 Ablation Studies and Design Choices

During model development, we conducted extensive ablation studies to identify the most effective architectural components for our task. We evaluated various graph convolutional layers from the `torch_geometric.nn` library, including `NNConv`, `GCNConv`, `GATv2Conv`, and `GraphConv`. Among these, the `TransformerConv` layer consistently achieved the best performance, likely due to its ability to incorporate edge features directly into the attention mechanism and its use of multi-head attention, which captures complex relational patterns between nodes more effectively.

In terms of loss functions, we experimented with a range of objectives, including regression losses, cross-entropy loss, CORN loss (Shi et al., 2023), and direct optimization of the quadratic weighted kappa (QWK) metric. Our final setup combines CORAL loss (Cao et al., 2020) with the weighted QWK loss (de La Torre et al., 2018), yielding improved convergence and performance. This hybrid objective leverages the ordinal-aware structure of CORAL while directly aligning training with the evaluation metric through QWK.

These empirical findings guided our final model design. We recommend using the `TransformerConv` layer in conjunction with a CORAL + QWK loss for tasks involving graph-based ordinal classification.

GNNinjas at BAREC Shared Task 2025: Lexicon-Enriched Graph Modeling for Arabic Document Readability Prediction

Passant Elchafei[*]
Ulm University, Germany
passant.elchafei@uni-ulm.de

Mayar Osama[*]
German University in Cairo, Egypt
mayar.osama@guc.edu.eg

Mohamed Rageh
German University in Cairo, Egypt
mohamad.rageh@student.guc.edu.eg

Mervat Abuelkheir
German University in Cairo, Egypt
mervat.abuelkheir@guc.edu.eg

Abstract

We present a graph-based approach enriched with lexicons to predict document-level readability in Arabic, developed as part of the Constrained Track of the BAREC Shared Task 2025. Our system models each document as a sentence-level graph, where nodes represent sentences and lemmas, and edges capture linguistic relationships such as lexical co-occurrence and class membership. Sentence nodes are enriched with features from the SAMER lexicon as well as contextual embeddings from the Arabic transformer model. The graph neural network (GNN) and transformer sentence encoder are trained as two independent branches, and their predictions are combined via late fusion at inference. For document-level prediction, sentence-level outputs are aggregated using max pooling to reflect the most difficult sentence. Experimental results show that this hybrid method outperforms standalone GNN or transformer branches across multiple readability metrics. Overall, the findings highlight that fusion offers advantages at the document level, but the GNN-only approach remains stronger for precise prediction of sentence-level readability.

1 Introduction

Accurately assessing the readability of Arabic documents is essential for educational technologies, language learning platforms, and adaptive content delivery systems. The task poses significant linguistic challenges due to the diglossic nature of Arabic, rich morphology, and the scarcity of large-scale annotated corpora (Imperial and Kochmar, 2023). The BAREC Shared Task 2025 (Elmadani et al., 2025a) addresses this by providing a fine-grained classification benchmark: assigning one of 19 readability levels to Arabic texts at both the sentence and document level.

Previous work on Arabic NLP has applied deep contextual models such as BERT variants for various classification tasks, including readability prediction (Al-Tamimi et al., 2014; Antoun et al., 2020). Although effective, these approaches typically operate only on text sequences and often overlook explicit structural and lexical relationships that can influence readability. In contrast, graph-based methods make it possible to encode document-level structure and linguistic relationships directly (Sun et al., 2023). In this work, we explicitly incorporate such relationships by leveraging the SAMER lexicon for lexical difficulty features and constructing a heterogeneous sentence-lemma graph with multiple edge types (e.g., HAS_LEMMA, OCCUR_WITH, IN_CLASS, IN_DOMAIN). This allows our model to combine the strengths of contextual embeddings with explicit lexical and structural graph modeling, which we show experimentally to improve both sentence-level and document-level readability prediction.

We propose a hybrid approach that represents each document as a graph, where nodes correspond to sentences and lemmas, and edges represent linguistic relationships such as `HAS_LEMMA`, `OCCUR_WITH`, and `IN_CLASS`. Each sentence node is enriched with difficulty signals from the SAMER lexicon (Al Khalil et al., 2020) and contextual sentence embeddings from the `readability-arabertv2-d3tok-CE` model, a fine-tuned variant of AraBERTv2 optimized for Arabic readability classification (Antoun et al., 2020).

To integrate both modalities, we train the GNN (graph modality) and the transformer (text modality) independently and use late fusion to merge their readability predictions at the end of inference. This approach combines the strengths of structured lexical-graph features and contextual

[*] Equal contribution.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 261–265
November 8-9, 2025 ©2025 Association for Computational Linguistics

text embeddings, without mixing intermediate features. Document-level labels are then obtained by pooling the sentence-level predictions, using max-pooling to reflect the most difficult sentence.

Our experiments demonstrate that this lexicon-enriched, confidence-aware, graph-based approach significantly improves prediction performance over individual branches. The results emphasize the importance of combining structured lexical knowledge with neural contextualization and fusion to better capture Arabic document readability.

2 Related Work

Automatic readability assessment has become a key area in NLP due to its applications in education, text simplification, and adaptive content delivery. In English, early studies relied on surface-level features such as sentence length and word frequency, followed by statistical models and, more recently, neural methods that capture semantic and discourse-level information (Imperial and Kochmar, 2023).

For Arabic, early research was constrained by resource scarcity and linguistic complexity. One of the first efforts was the AARI index (Al-Tamimi et al., 2014), which used handcrafted lexical and syntactic features derived from academic curricula. Later, the SAMER Lexicon (Al Khalil et al., 2020) introduced a large-scale graded vocabulary resource. Subsequently, it was showcased in a word-level readability visualization system designed for assisted text simplification (Hazim et al., 2022). More recently, the SAMER Corpus (Alhafni et al., 2024) provided the first manually annotated Arabic parallel dataset for text simplification targeting school-aged learners. These resources provided the foundation for subsequent work.

In recent years, several datasets have advanced Arabic readability modeling. The BAREC corpus (Elmadani et al., 2025b) provides a large-scale benchmark with 19 readability levels at both the sentence and document level, while the DARES dataset (El-Haj et al., 2024) focuses on Saudi school textbooks. Complementary approaches, such as AraEyebility (Baazeem et al., 2025), integrate eye-tracking signals to connect human cognitive processing with readability prediction. In addition, (Liberato et al., 2024) explored strategies for Arabic readability modeling, highlighting the need to combine lexical resources with modern learning-based approaches. A survey by (Cavalli-Sforza et al., 2018) provides an overview of the challenges and future directions for Arabic readability assessment.

Overall, most Arabic readability models have focused on surface features or contextual embeddings in isolation, with limited integration of structured lexical knowledge. To our knowledge, no prior work has combined lexicon-enrichment with graph-based modeling for Arabic document readability. Our work addresses this gap by integrating the SAMER Lexicon into a heterogeneous sentence-lemma graph, capturing both vocabulary difficulty and structural relations to improve fine-grained readability prediction.

3 System Overview

The purpose of our approach is to capture the linguistic characteristics and the relationships between the features of two datasets: BAREC (Elmadani et al., 2025b) and SAMER (Al Khalil et al., 2020). The BAREC dataset consists of **sentences** annotated with their corresponding **readability levels**. The SAMER dataset consists of **lemmas**, each associated with an average **readability level** across different dialects, along with additional features such as frequency of occurrence and part-of-speech (POS) tags for each *(lemma, readability level)* pair.

We integrate the two datasets by extracting lemmas from the sentences while preserving their POS tags and recording the count of diacritics. The extraction of lemmas was performed using the CAMeL Tools Morphology Analyzer (Obeid et al., 2020). Each extracted lemma is then matched against the SAMER lexicon to enrich it with statistical attributes such as average readability, frequency, and POS. This alignment ensures that the SAMER lexicon contributes directly to the graph as node features rather than as isolated entries.

The combined data is reformulated into a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of multiple node and edge types. The node set $\mathcal{V}$ includes:

- **Sentences**: Represented by 768-dimensional embeddings obtained from the CAMeL-Lab Arabic readability model (readability-arabertv2-d3tok-CE), augmented with linguistic features.
- **Lemmas**: Characterized by statistical attributes such as average readability and frequency.

- **Classes**: Educational difficulty levels, one-hot encoded as Foundational, Advanced, or Specialized.
- **Domains**: Subject domains encoded as Arts & Humanities, STEM, or Social Sciences.

The main objective of our approach is to leverage both classical linguistic features from the two datasets and deep Graph Neural Networks (GNNs) to capture hidden patterns within the data. The edge set $\mathcal{E}$ contains the following directed relations:

- `sentence` $\rightarrow$ `lemma` (*HAS_LEMMA*): Indicates lexical composition.
- `lemma` $\leftrightarrow$ `lemma` (*OCCUR_WITH*): Represents lemma co-occurrence in context.
- `sentence` $\rightarrow$ `class` (*IN_CLASS*): Connects each sentence to its labeled difficulty class.
- `sentence` $\rightarrow$ `domain` (*IN_DOMAIN*): Links sentences to their broader academic domain.

Once the data is structured into the graph format, the first step is to apply input feature transformation, where we transform the node features into the model's hidden dimensions, for which we used a linear layer between the original dimensions to the target dimension to find optimal projection.

$$h_v^{(0)} = W_{\text{in}}^{(\tau)} x_v, \quad \text{for } v \in \mathcal{V}_\tau$$

where $h_v^{(0)}$ is the initial hidden representation of node v after projection, $W_{\text{in}}^{(\tau)}$ is the trainable weight matrix for input transformation for node type, $\mathcal{V}_\tau$ denotes nodes of type τ, and x_v is the raw feature vector.

The core of the model consists of a stack of SAGEConv (Hamilton et al., 2017) hidden layers. Each is used to learn the graph embeddings over the heterogeneous graph. It uses neighbor sampling and aggregation. Each layer applies a learnable linear transformation to the combined features; this transformation allows the model to learn complex feature interaction while maintaining consistent dimensions across the layers.

The model consistes of 4 GNN layers, for which we use ReLU activation function σ and layer normalization to avoid linearity and improve the gradient flow.

$$h_v^{(k)} = \sigma\left(\text{AGGREGATE}_{\text{type}}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}_{\text{type}}(v)\right\}\right)\right)$$

where $h_v^{(k)}$is the hidden representation of node v at layer k, $h_u^{(k-1)}$ is the hidden representation of neighbor node u from the previous layer, and $\mathcal{N}_{\text{type}}(v)$ denotes the set of neighboring nodes of v connected via a specific edge type. Additionally, we use a residual connection per layer. This preserves the features and provides more stable training, especially for the sentence nodes.

$$h_v^{(k)} \leftarrow \text{LayerNorm}\left(h_v^{(k)} + h_v^{(k-1)}\right)$$

Finally, an MLP layer used for the classification.

$$y_v = \text{MLP}(h_v^{(L)})$$

4 Experimental Results

We conduct experiments on both sentence-level and document-level readability prediction tasks, as defined in the BAREC Shared Task 2025. For sentence-level classification, each sentence is represented as a node in the graph and labeled with one of 19 readability levels. For document-level prediction, we reuse the same model architecture and apply aggregation over sentence-level predictions. Specifically, we take the most difficult predicted sentence level (i.e., max pooling) as the document's predicted readability level based on the intuition that the most complex sentence may determine the document's comprehensibility floor.

We evaluate two configurations:

- **Late Fusion:** Combining weighted outputs from the GNN and transformer-based sentence encoder.
- **GNN Only:** Using the graph-based model without fusion.

The results in Table 1 show distinct trends between sentence-level and document-level tasks. For document-level prediction, Late Fusion outperforms the GNN-only baseline in both Quadratic Weighted Kappa (QWK; 76.9% vs. 75.6%) and exact accuracy (42.0% vs. 40.0%), while maintaining similar scores in the other metrics. QWK is a standard evaluation metric for ordinal classification that accounts for the degree of disagreement between predicted and true labels, making it particularly relevant for readability level prediction.

In contrast, for sentence-level prediction, the GNN-only model achieves substantially higher accuracy (50.0% vs. 41.4%) and better results in most metrics, despite both models having the same

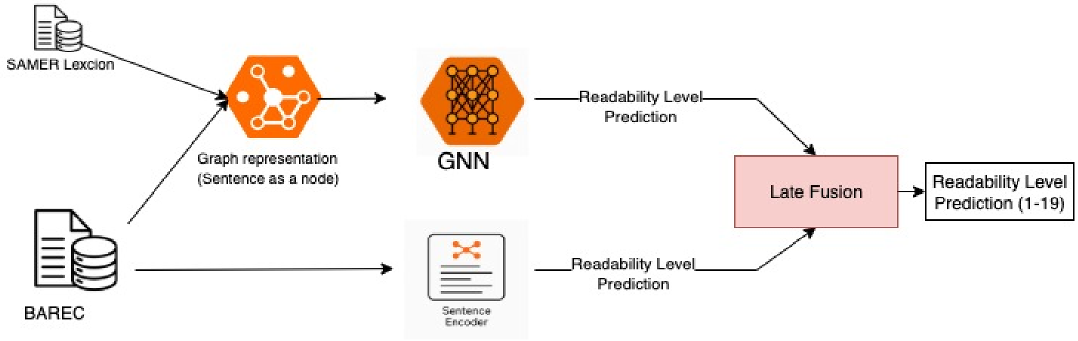


Figure 1: Overview of our proposed hybrid architecture for Arabic readability prediction. Sentence-level graphs are constructed using lexical relations from the SAMER lexicon and structural information from BAREC data. The GNN branch processes the graph to produce a softmax probability distribution over 19 readability levels, while the sentence encoder branch generates parallel probabilities from contextual embeddings. Inference uses late fusion, where both probability vectors are combined at the prediction stage using a tunable weight, yielding the final readability level for each sentence or aggregated document.

Task Level	Model Variant	QWK	Acc	Acc +/-1	Dist	Acc 7	Acc 5	Acc 3
Document-Level	GNN Only	75.6	40.0	83.0	0.8	60.0	60.0	90.0
Document-Level	Late Fusion	**76.9**	**42.0**	82.0	0.8	60.0	61.0	90.0
Sentence-Level	GNN Only	**78.5**	**50.0**	67.2	1.3	61.2	66.1	74.9
Sentence-Level	Late Fusion	**78.5**	41.4	65.9	1.4	55.4	62.6	72.7

Table 1: Performance of the GNN-based model and Late Fusion on sentence-level and document-level readability prediction, evaluated with Quadratic Weighted Kappa (QWK), accuracy, accuracy within ±1, distribution score, and accuracy at multiple granularity levels (7, 5, and 3).

QWK (78.5%). This indicates that, at the finer sentence granularity, the graph-based model alone is more effective, while the fusion approach may dilute some of the GNN's discriminative power for exact classification.

Overall, the findings highlight that fusion offers advantages at the document level, but the GNN-only approach remains stronger for precise sentence-level readability prediction.

5 Conclusion

In this paper, we proposed a hybrid approach for Arabic document readability prediction by combining graph-based reasoning with contextual transformer-based modeling. Our architecture integrates lexical difficulty knowledge from the SAMER lexicon, sentence embeddings from a fine-tuned AraBERTv2 variant, and a structured graph representation of each document.

For sentence-level prediction, we demonstrated the benefits of lexicon-enriched heterogeneous graph modeling using a weighted GNN. For document-level prediction, we reuse the same graph setup and infer the document's label by selecting the maximum difficulty among sentence-level predictions. This design aligns with the task's objective of identifying the highest comprehension barrier within a document.

By applying late fusion between the GNN and transformer predictions, we achieved stronger performance across both levels. Our results highlight the complementary nature of structural and contextual signals and the promise of fusion-based systems for fine-grained Arabic readability tasks.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Abdel-Karim Al-Tamimi, Manar Jaradat, Nuha Aljarrah, and Sahar Ghanim. 2014. Aari: Automatic arabic readability index. *International Arab Journal of Information Technology*, 11:370–378.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language*

Resources and Evaluation (LREC-COLING 2024), pages 16079–16093, Torino, Italia. ELRA and ICCL.

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.

Ibtehal Baazeem, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2025. Araeyebility: Eye-tracking data for arabic text readability. *Computation*, 13(5).

Violetta Cavalli-Sforza, Hind Saddiki, and Naoual Nassiri. 2018. Arabic readability research: Current state and future directions. *Procedia Computer Science*, 142:38–49. Arabic Computational Linguistics.

Mo El-Haj, Sultan Almujaiwel, Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2024. DARES: Dataset for Arabic readability estimation of school materials. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 103–113, Torino, Italia. ELRA and ICCL.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, pages 1024–1034. Curran Associates, Inc.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Joseph Marvin Imperial and Ekaterina Kochmar. 2023. Automatic readability assessment for closely related languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5371–5386, Toronto, Canada. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. 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 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Qi Sun, Kun Zhang, Kun Huang, Tiancheng Xu, Xun Li, and Yaodi Liu. 2023. Document-level relation extraction with two-stage dynamic graph attention networks. *Knowledge-Based Systems*, 267:110428.

ZAI at BAREC Shared Task 2025: AraBERT CORAL for Fine Grained Arabic Readability

Ahmad M. Nazzal
ZAI Arabic Language Research Center / Zayed University

Email: Ahmad.Nazzal@zu.ac.ae

Abstract

Readability assessment is essential for effective communication of scientific and medical content in Arabic. We present a system for the BAREC 2025 Shared Task Arabic Readability Assessment. The system fine-tunes AraBERTv2 with a CORAL ordinal head, applies AraBERT-specific preprocessing, and selects checkpoints using Quadratic-Weighted Kappa (QWK) with early stopping. Our model achieves a QWK of 85.5 on the Sentence Blind Test, demonstrating its effectiveness for automatic Arabic readability prediction.

1 Introduction

Automatic readability assessment supports education, accessibility, and editorial workflows by estimating how difficult a text is for a target audience (Hazim et al., 2022; Liberato et al., 2024). For Arabic, this task is especially challenging: the language is morphologically rich and it exhibits diglossia between Modern Standard Arabic and regional dialects, which complicates lexical and morphosyntactic cues used by models (Asadi and Abu-Rabia, 2019; Ferguson, 1959; Saiegh-Haddad and Ghawi-Dakwar, 2017; Taha and Saiegh-Haddad, 2016). The BAREC 2025 shared task addresses these challenges with a large, fine-grained sentence-level benchmark annotated into 19 ordered levels, accompanied by clear guidelines and an evaluation based on quadratic-weighted kappa (QWK) (Al Khalil et al., 2020; Elmadani et al., 2025a,b; Habash et al., 2025). We participate in the Sentence-level (Strict) track. We present a compact, reproducible system: AraBERTv2 (Antoun et al., 2020) with a rank-consistent ordinal regression (CORAL) head (Aicher et al., 2022; Cao et al., 2020). Training uses early stopping with QWK-based model selection; inference applies a single development-tuned threshold to convert cumulative probabilities into one of the 19 levels (no temperature scaling). This simple architecture achieves strong performance on the Blind Test.

2 Background

Early work estimated readability using surface proxies such as sentence/word length and syllabification, yielding indices like Flesch Reading Ease and Dale–Chall (Dale and Chall, 1948; Flesch, 1948). Contemporary approaches treat readability as supervised prediction over lexical, syntactic, and distributional features, increasingly with pretrained language models. Fine-grained readability labels are ordered; modeling them as nominal classes discards rank information. Ordinal regression methods—especially CORAL, which learns $K-1$ binary thresholds for events $y > k$—enforce label order and are often preferable to softmax for such targets. As resources, the BAREC benchmark provides a large sentence-level corpus with 19 readability levels, detailed annotation guidelines, and official splits (*Open Dev*, *Open Test*, *Blind Test*) tailored for shared-task evaluation (Elmadani et al., 2025a,b; Habash et al., 2025). Related Arabic resources such as SAMER target text simplification rather than graded readability but reflect a broader interest in accessibility for Arabic texts (Alhafni et al., 2024; Al Khalil et al., 2020). Given these factors—Arabic's linguistic properties, the ordinal nature of labels, and QWK as the official metric—pretrained Arabic encoders such as AraBERT offer a natural foundation for readability systems; we therefore build on AraBERTv2 and an ordinal head in the methods that follow (Antoun et al., 2020).

3 System Overview

3.1 Encoder and Preprocessing

We Fine-tune aubmindlab/bert-base-arabertv2. We start by normalizing/segmented sentences with the AraBERTPreprocessor and tokenized with fixed max length = 128 (no dynamic padding).

3.2 Ordinal Head (CORAL)

Let $K = 19$ and labels $y \in \{0, \ldots, K-1\}$ (with labels shifted by -1 during training). The ordinal head outputs logits z_k for $k = 1, \ldots, K-1$; after applying the sigmoid function, $\sigma(z_k) \approx P(y > k)$. The training targets are defined as cumulative indicators

$$t_k = \begin{cases} 1, & y > k \\ 0, & \text{otherwise} \end{cases}$$

The loss function is the binary cross-entropy with logits, summed over all thresholds:

$$L = \sum_{k=1}^{K-1} \text{BCEWithLogits}(z_k, t_k),$$

thereby enforcing consistent ordering of the predicted categories (Cao et al., 2020).

3.3 Training and Selection

Optimization uses AdamW, a linear schedule with warmup, gradient clipping, label smoothing $= 0.0$ (smoothing hurt this fine-grained ordinal task), and early stopping (patience $= 2$). We set `metric_for_best_model = eval_qwk` and `greater_is_better = True` so the saved model maximizes QWK.

3.4 Inference (Single Threshold Only)

We convert the $(K-1)$ probabilities to a level by counting how many exceed a single threshold t, tuned on the dev set to maximize QWK. No temperature scaling or additional calibration is used in the final system.

4 Experimental Setup

We use the organizers' *Open Train*, *Open Dev*, *Open Test*, and *Blind Test* (sentence track). Splits are unchanged. Key hyperparameters: Encoder: AraBERTv2; max length $=$ 128. Optimizer: AdamW (lr $= 2 \times 10^{-5}$), weight decay $= 0.01$; linear decay; warmup $= 6$; max-grad-norm $= 1.0$. Batching: train batch size $=$ 16, eval batch size $= 32$; gradient accumulation $= 2$. Regularization: label smoothing $=$ 0.0; early stopping patience $= 2$. Precision: FP16 on T4 (BF16 if available). Selection: best checkpoint by eval_qwk; threshold t tuned on dev. Implementation: Transformers 4.54.0; Datasets ≥ 2.18.

Data	QWK	Acc. (%)	Acc. ±1 (%)	Dist.	Acc. 7	Acc. 5	Acc. 3
Open Dev.	66.6	41.5	55.8	1.6	51.9	57.4	64.9
Open Test	72.9	46.6	61.2	1.4	55.8	61.0	69.3
Blind Test	85.5	35.6	74.2	1.0	64.4	69.3	75.8

Table 1: Model performance across datasets. QWK = quadratic weighted kappa.

5 Results

On the Open Dev set, the model reached a QWK of 66.6; performance improved on the Open Test set (72.9) and peaked on the Blind Test set (85.5). Accuracy was moderate overall (35–47%), but accuracy within one level was substantially higher (56–74%), indicating the model captures ordinal trends even if exact prediction is difficult. The distribution score decreased from 1.6 on Dev to 1.0 on Blind, suggesting better calibration on held-out data. As expected, replacing CORAL with a nominal softmax head reduced QWK, confirming the benefit of enforcing label order. Label smoothing and temperature scaling both impaired dev QWK, so the final system uses neither. No explicit error analysis was conducted. Results are summarized in Table 1.

6 Conclusion

We presented a compact system for fine-grained Arabic readability in the BAREC 2025 shared task. The method combines AraBERTv2 with a CORAL ordinal head, trains with QWK-based model selection and early stopping and predicts with a single development-tuned threshold. Without external data or complex ensembling, the system achieves QWK = 85.5 on the Sentence Blind Test. The results support two takeaways: (i) respecting label order via an ordinal head is effective for 19-level readability; and (ii) aligning selection and postprocessing with the official metric (QWK) is a simple, high-leverage choice.

Limitations

We rely solely on the shared task splits, no external corpora or augmentation. Domain transfer beyond BAREC is untested. A single encoder is used; larger backbones or multilingual pretraining were not explored due to time/compute. No document-level context or explicit linguistic features (e.g., morphological complexity, type–token ratio) are used. We focus on core choices (ordinal vs. nominal, smoothing, temperature). Future work includes error analysis, and exploring alternative ordinal objectives (e.g., CORN) and document-level context.

Ethics Statement

The author declares an affiliation with an institution that contributed to the preparation of the shared task. None of the organizers contributed to the conception, development, or evaluation of our systems. All information and resources used were based exclusively on resources publicly released to all participants, without any form of privileged access or guidance.

References

Annalena Aicher, Alisa Gazizullina, Aleksei Gusev, Yuri Matveev, and Wolfgang Minker. 2022. Towards speech-only opinion-level sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2000–2006, Marseille, France. European Language Resources Association.

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

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.

Ibrahim A. Asadi and Salim Abu-Rabia. 2019. The impact of the position of phonemes and lexical status on phonological awareness in the diglossic arabic language. *Journal of Psycholinguistic Research*, 48(5):1051–1062.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331.

Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Charles A. Ferguson. 1959. Diglossia. *WORD*, 15(2):325–340.

Rudolf. Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Elinor Saiegh-Haddad and Ola Ghawi-Dakwar. 2017. Impact of diglossia on word and non-word repetition among language impaired and typically developing arabic native speaking children. *Frontiers in Psychology*, 8:2010.

Haitham Taha and Elinor Saiegh-Haddad. 2016. The role of phonological versus morphological skills in the development of arabic spelling: An intervention study. *Journal of Psycholinguistic Research*, 45(3):507–535.

ANLPers at BAREC Shared Task 2025: Readability of Embeddings: Training Neural Readability Classifiers on the BAREC Corpus

Serry Sibaee[1*] **Yasser Alhabashi**[1] **Omer Nacar**[2] **Adel Ammar**[1] **Wadii Boulila**[1]
[1]Prince Sultan University, Riyadh, Saudi Arabia
[2]Tuwaiq Academy – Tuwaiq Research and Development Center
{yalhabashi, ssibaee, aammar , wboulila}@psu.edu.sa
{o.najar}@tuwaiq.edu.sa
[*]Corresponding author: ssibaee@psu.edu.sa

Abstract

This paper presents a neural approach to Arabic readability assessment using the BAREC corpus for fine-grained classification across 19 readability levels. Our two-stage system combines embeddings from multiple pre-trained Arabic transformer models (ARBERTv2, MARBERTv2, AraBERT) with a Multi-Layer Perceptron classifier. We achieve competitive performance with Quadratic Weighted Kappa scores of 73.00-76.35, accuracy of 44.73%, and adjacent accuracy of 61.40%, within 8% of baseline models. The system offers significant practical advantages including rapid training time (10 minutes per experiment), compact architecture (12-15 million parameters), and efficient inference, making it suitable for resource-constrained deployment. Our analysis identifies dataset quality challenges including inconsistent diacritization and annotation issues that impact performance. This work provides a foundation for practical Arabic readability assessment tools in educational applications.

1 Introduction

Automatic readability assessment has become increasingly important in educational technology, content adaptation, and accessibility applications in many languages including Arabic (Liberato et al., 2024). Traditional readability metrics rely heavily on surface-level features such as sentence length and syllable counts (Uçar et al., 2024), which often fail to capture the nuanced linguistic complexity that affects human comprehension. Recent advances in neural language models and contextual embeddings offer new opportunities to develop more sophisticated readability classifiers that can better model the relationship between text characteristics and reading difficulty (Hazim et al., 2022).

This work investigates the application of modern neural architectures and embedding techniques to readability classification using the BAREC corpus. We address key challenges in current modeling approaches including the need for better representation of semantic complexity, syntactic structures, and discourse coherence. Our novel approach combines multiple embedding strategies with attention mechanisms to create interpretable readability predictions. The contributions of this work include empirical analysis of embedding effectiveness for readability tasks and a comprehensive evaluation framework for neural readability classifiers.

2 Background

Text readability plays a vital role in ensuring comprehension, retention, and engagement, especially in educational and medical (Venturi et al., 2015) contexts where aligning reading material with student proficiency is critical. Fine-grained readability frameworks, such as Fountas and Pinnell (Ransford-Kaldon et al., 2010) for English and the 19-level system for Arabic (Elmadani et al., 2025b), and some researchers used RL to develop readability assessment systems (Mohammadi et al., 2023) are widely used to support literacy development.

In this work, we participate in the **BAREC Shared Task 2025 on Arabic Readability Assessment:Sentence-level-Open** (Elmadani et al., 2025a), which focuses on sentence-level classification into one of 19 Taha-Thomure levels.(Taha-Thomure, 2017), from kindergarten to postgraduate proficiency. We use the newly released **BAREC corpus** (Elmadani et al., 2025b), a large, balanced dataset (splitted as: 54845 training sample, 7310 validation, 7286 test and 3420 blind-test) annotated according to the fine-grained guidelines outlined by (Habash et al., 2025). The task is a challenging multi-class classification problem requiring precise sentence-level prediction.

The corpus is derived in part from the **SAMER Arabic Text Simplification Corpus** (Alhafni et al., 2025) and (Al Khalil et al., 2020), and Figure 1

Proceedings of The Third Arabic Natural Language Processing Conference, pages 269–273
November 8-9, 2025 ©2025 Association for Computational Linguistics

illustrates the distribution of sentences across the 19 levels. Our system aims to automatically predict the correct level for each input sentence from raw Arabic text, enabling more effective support for educational applications and adaptive reading technologies.

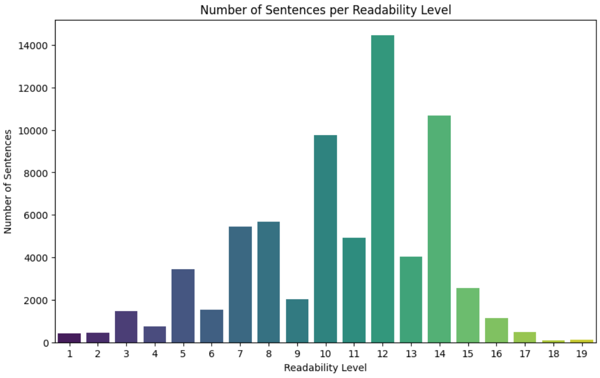


Figure 1: Distribution of sentences across the 19 readability levels in the BAREC corpus

Below are example sentences (randomly selected from the dataset from each level) along with their assigned readability levels:

- Level 1: ضعيفٌ - "Weak"
- Level 2: مرحباً. - "Hello."
- Level 3: الْقُرْآنُ الْكَرِيمُ - "The Noble Quran"
- Level 4: الْحَمْدُ لله - "Praise be to Allah"
- Level 5: أُصدرُ حُكْمًا: - "I issue a judgment:"
- Level 6: قلت الأمل - "I said hope"
- Level 7: أرجوك تخلص فورًا من … الخ - "Please get rid of immediately... etc."
- Level 8: أَتَأَمَل المَشْهَدَ الآتِيَ، ثُمَ … الخ - "I contemplate the following scene, then... etc."
- Level 9: (كيف أهدأ وأنا بالأمس … الخ - "(How can I calm down when yesterday... etc."
- Level 10: ثالثًا: سريعة جدًا، تتحرك … الخ - "Third: very fast, it moves... etc."
- Level 11: لست متأكدًا أي أفق … الخ - "I’m not sure which horizon... etc."
- Level 12: وأشهر مدنه: أرجيش، بَدْلِيس أو … الخ - "And its most famous cities: Erciş, Bitlis or... etc."
- Level 13: فإذا جاءتْ بشدّتها وغمراتِها، عندَ … الخ - "When it comes with its intensity and overwhelming force, when... etc."
- Level 14: وَقَالَ مَعْهَدُ بُحُوثِ السَرطَانِ … الخ - "And the Cancer Research Institute said... etc."
- Level 15: ويَنطوي هذا التَوْزيع في الوظائف … الخ - "And this distribution in functions involves... etc."
- Level 16: ولكن قيمتَه لا تُقدَرُ في … الخ - "But its value cannot be estimated in... etc."
- Level 17: تَسمَعُ للحَليِ وَسْوَاساً إذا … الخ - "You hear a whisper of jewelry when... etc."
- Level 18: تَرَى اللَحزَ الشَحيحِ إِذَا … الخ - "You see the meager flesh when... etc."
- Level 19: يَتْبَعْنَ قُلَةَ رَأْسِهِ وَكَأَنَهُ … الخ - "They follow the crown of his head as if he... etc."

The complexity progression across readability levels is also reflected in the sentence length characteristics. Figure 2 demonstrates the distribution of word counts per sentence across different readability levels, showing how sentence complexity generally increases with higher readability levels.

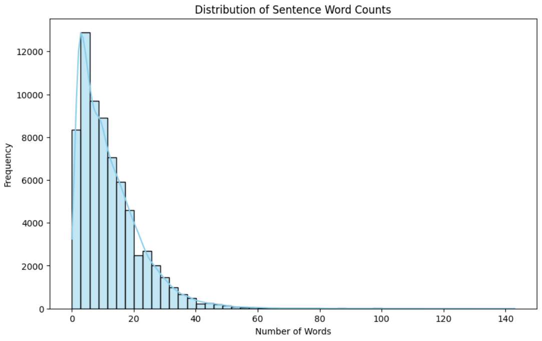


Figure 2: Distribution of word counts per sentence

Developing accurate automatic readability assessment models for Arabic is essential for advancing literacy education, supporting language learning applications, and improving academic performance evaluation. This task plays a vital role in standardizing Arabic text complexity assessment and contributes to the broader goal of enhancing Arabic language education through technology-driven tools.

3 System Overview

Our system for automatic readability assessment is a two-stage pipeline designed to first extract deep linguistic features from Arabic text and then predict a readability score (Sibaee et al., 2024). This architecture addresses the core challenge of capturing the complex interplay of semantic and syntactic features that determine text difficulty[1].

The entire process can be conceptualized as a composition of two functions. First, an embedding function, E, maps the input text T to a fixed-size vector representation $\mathbf{e}$. This vector is then processed by a prediction model, M, parameterized by trainable weights W, to produce the final readability score, $\hat{y}$.

1. **Text Embedding:** $\mathbf{e} = E(T)$, where $\mathbf{e} \in \mathbb{R}^d$
2. **Readability Prediction:** $\hat{y} = M_W(\mathbf{e})$

Stage 1: Multi-Model Text Embedding (E). To create a robust feature vector, we generate embeddings from an ensemble of pre-trained Arabic transformer models: ARBERT, AraBERT, and MARBERT. Our design decision to use multiple models is to ensure the final representation is rich and generalized. For each model, the input text is tokenized, and the model outputs contextualized embeddings for every token. We compute a single sentence-level vector for each model by taking the mean of its token output embeddings. The final embedding, $\mathbf{e}$, is the element-wise average of the vectors from all three models. This averaging technique smooths the representation space and captures a broader range of linguistic nuances critical for readability assessment.

Stage 2: Readability Prediction Model (M). The resulting embedding vector $\mathbf{e}$ serves as the input to our prediction model, M, which is a Multi-Layer Perceptron (MLP). This feed-forward neural network is configured with several hidden layers and is trained to learn the complex, non-linear mapping from the dense text features to a continuous readability score. The model's parameters, W, are optimized using a regression loss function to minimize the error between its predicted scores and the ground-truth labels.

4 Experimental Setup

4.1 Dataset and Preprocessing

We evaluate our approach on the CAMeL-Lab/BAREC-Shared-Task-2025-sent dataset (Elmadani et al., 2025a) from Hugging Face (we did not evaluate on the validation split so we added them to the training to expand the samples)[2], a benchmark for Arabic readability assessment. The preprocessing pipeline consists of two steps: (1) text normalization by removing non-Arabic letters and numbers, and (2) lemmatization using Sina Tools (Hammouda et al., 2024) to reduce text nosiness and data sparsity. The methodology consist of trying multiple combination of the pre-processing techniques in the expirements which showed a very closed results either with them or direct training without pre-processing.

4.2 Model Architecture

Our system combines pre-trained embedding models with a Multi-Layer Perceptron (MLP) classifier, implemented in PyTorch using Hugging Face libraries. We evaluate two embedding categories: general multilingual models (LaBSE (Reimers and Gurevych, 2020), all-MiniLM-L6-v2, Matryoshka-based (Nacar et al., 2025)) and Arabic-specific BERT models (ARBERTv2, MARBERTv2 (Abdul-Mageed et al., 2021), AraBERTv2 (Antoun et al.)).

4.3 Training Configuration

The MLP architecture uses 3-4 hidden layers in descending configuration (e.g., [4096, 2048, 1024, 512]). Training employs AdamW optimizer with learning rates of 10^{-4} or 10^{-5}, batch sizes up to 65,536 (using A100-80GB), and 800-2000 epochs with early stopping. Regularization includes dropout (0.3-0.5) and weight decay (10^{-5}). All experiments use random seed 42 for reproducibility.

5 Results

We conducted extensive experiments across multiple configurations, achieving consistent performance on key metrics (QWK, Accuracy, Adjacent Accuracy) with QWK scores ranging from 65 to 76. This section presents our most promising results on both test and blind-test datasets provided by the shared task.

5.1 Experimental Configurations

After conducting numerous experiments, we observed that the results were highly similar; therefore, we selected the two best configurations, tak-

[1] The system is open-sources on github https://github.com/riotu-lab/readability_library_training

[2] note:The system is not directly comparable to other participants' systems because it uses the development set for training.

ing into account their differences in specific aspects, as shown in Table 1.

Parameter	Exp-1	Exp-2
Emb. Model	ARBERTv2	MARBERTv2
Input Size	768	768
Hidden Layers	SY	[SY, 512]
Dropout Rate	0.2	0.4
Learning Rate	10^{-5}	10^{-2}
Epochs	800	1200
Weight Decay	10^{-5}	$3 * 10^{-5}$
Early Stop	25	100
Scheduler	5	25

Table 1: Training configurations for best-performing experiments. Note: the default hidden layer is [4096, 2048, 1024] symbolized as 'SY'

5.2 Main Results

The primary findings of our experiments are presented in Table 2, which provides a comparative overview of model performance across different evaluation settings. The results indicate that both experiments achieved nearly identical accuracy and adjusted accuracy, with only slight variations in QWK. This consistency demonstrates the robustness of the approach across test and blind test datasets.

Exp.	Accuracy (%)	Adj Accuracy (%)	QWK
Exp-1	44.73	61.35	76.35
Exp-2	44.70	61.40	73.00

Table 2: Performance results on test dataset (exp-1) and blind test (exp-2)

5.3 Analysis and Discussion

Through extensive experimentation and dataset analysis (Sibaee et al., 2025), we identify two key observations:

5.3.1 Dataset Characteristics

Our analysis reveals several data quality issues that impact model performance: (1) inconsistent word diacritization across texts, (2) irregular punctuation usage patterns, (3) incomplete or fragmented sentences containing irrelevant symbols and noise, and (4) incorrect readability classifications for certain sentence types, particularly poetry verses and literary excerpts. These inconsistencies introduce noise that affects the reliability of readability predictions[3].

5.3.2 Model Architecture Performance

While our approach did not achieve state-of-the-art results, it demonstrates competitive performance compared to the baseline model (Elmadani et al., 2025b), achieving QWK scores within 8% of the baseline. However, our pipeline offers significant practical advantages: (1) substantially faster training time (approximately 10 minutes per experiment), (2) compact model size (12-15 million parameters). These characteristics make our approach particularly suitable for fast training in resource-constrained.

6 Conclusion

This research demonstrates that efficient neural architectures can achieve competitive performance for Arabic readability assessment while offering substantial practical advantages. Our two-stage system achieved QWK scores of 73.00-76.35 on the BAREC corpus, performing within 8% of baseline models with significantly faster training time and compact model size. The approach successfully addresses deployment considerations critical for educational technology applications in resource-constrained environments. Our analysis identified important dataset quality issues including inconsistent diacritization and annotation challenges that affect model performance. While not achieving state-of-the-art results, this work establishes a practical foundation for Arabic readability classification and highlights key areas for future corpus development and model improvement.

Acknowledgments

We would like to thank Prince Sultan University for their generous support in enabling this 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.

[3] Also as shown in figure 1, there is small amount of high level sentences so we expanded it using more Arabic poems and some teaching manzomat (more than 13K sample) on the link `https://huggingface.co/datasets/JadwalAlmaa/Expand_BAREC`. Note: we did not used this new dataset in our training.

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2025. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Nizar Habash, Hanada Taha-Thomure, Khalid Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Tymaa Hammouda, Mustafa Jarrar, and Mohammed Khalilia. 2024. Sinatools: Open source toolkit for arabic natural language processing. *Preprint*, arXiv:2411.01523.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Hamid Mohammadi, Seyed Hossein Khasteh, Tahereh Firoozi, and Taha Samavati. 2023. Text as environment: A deep reinforcement learning text readability assessment model. *Preprint*, arXiv:1912.05957.

Omer Nacar, Anis Koubaa, Serry Sibaee, 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*.

Carolyn R Ransford-Kaldon, E Sutton Flynt, Cristin L Ross, Louis Franceschini, Todd Zoblotsky, Ying Huang, and Brenda Gallagher. 2010. Implementation of effective intervention: An empirical study to evaluate the efficacy of fountas & pinnell's leveled literacy intervention system (lli). 2009-2010. *Center for Research in Educational Policy (CREP)*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Serry Sibaee, Abdullah Alharbi, Samar Ahmad, Omer Nacar, Anis Koubaa, and Lahouari Ghouti. 2024. ASOS at KSAA-CAD 2024: One embedding is all you need for your dictionary. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 697–703, Bangkok, Thailand. Association for Computational Linguistics.

Serry Sibaee, Omer Nacar, Adel Ammar, Yasser Al-Habashi, Abdulrahman Al-Batati, and Wadii Boulila. 2025. From guidelines to practice: A new paradigm for arabic language model evaluation. *Preprint*, arXiv:2506.01920.

Hanada Taha-Thomure. 2017. *Arabic Language Text Leveling (معايير هنادا طه لتصنيف مستويات النصوص العربية)*. Educational Book House (دار الكتاب التربوي للنشر والتوزيع).

Suna-Şeyma Uçar, Itziar Aldabe, Nora Aranberri, and Ana Arruarte. 2024. Exploring automatic readability assessment for science documents within a multilingual educational context. *International Journal of Artificial Intelligence in Education*, 34(4):1417–1459.

Giulia Venturi, Tommaso Bellandi, Felice Dell'Orletta, and Simonetta Montemagni. 2015. NLP–based readability assessment of health–related texts: a case study on Italian informed consent forms. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 131–141, Lisbon, Portugal. Association for Computational Linguistics.

MarsadLab at BAREC Shared Task 2025: Strict-Track Readability Prediction with Specialized AraBERT Models on BAREC

Shimaa Ibrahim[1], Md. Rafiul Biswas[2], Mabrouka Bessghaier[1], Wajdi Zaghouani[1]
[1]Northwestern University in Qatar
[2]Hamad Bin Khalifa University (HBKU), Qatar
{shimaa.ibrahim,mabrouka.bessghaier,wajdi.zaghouani}@northwestern.edu
mbiswas@hbku.edu.qa

Abstract

The BAREC 2025 Shared Task on Arabic readability targets 19 levels of ordinal prediction at the sentence and document levels under strict training. This paper describes a two stages system that basically starts with BAREC-tuned AraBERT checkpoints and then specializes on the Strict splits with Weighted Kappa Loss (WKL), an objective aligned with Quadratic Weighted Kappa (QWK). A single architecture with inputs specific to each track is utilized for both tracks. On the Strict setting, our best systems reach 0.842/0.841 QWK (public/blind) at the sentence level and 0.828/0.790 QWK at the document level.

1 Introduction

Automatic readability assessment (ARA) estimates how difficult a text is for a target audience. For Arabic, the task is challenging due to morphological richness, orthographic variation, and the coexistence of Modern Standard Arabic (MSA) with regional dialects (Habash, 2010; Cavalli-Sforza et al., 2018). These factors complicate tokenization, feature extraction, and modeling, especially for rare ordinal labels, where small lexical or syntactic differences can shift a sentence between adjacent levels.

The BAREC 2025 Shared Task (Elmadani et al., 2025b) provides a large benchmark with 19 readability levels at the sentence and document levels, spanning multiple domains and genres. Companion resources include a corpus paper (Elmadani et al., 2025a) and detailed annotation guidelines (Habash et al., 2025). We focus on the Strict track, which constrains training to the official data only, resulting in limited data, class imbalance, and closely spaced ordinal labels—conditions that favor pretrained models and ordinal-aware objectives.

Earlier Arabic readability systems relied on manual indicators (e.g., sentence/word length, frequency, morphology) and classical ML, e.g., AARI and OSMAN (Al Tamimi et al., 2014; El-Haj and Rayson, 2016); surveys report that such features under-represent semantics and discourse (Cavalli-Sforza et al., 2018). With Arabic PLMs, performance improved across many tasks (e.g., AraBERT, MARBERT) (Antoun et al., 2020; Abdul-Mageed et al., 2021), but standard fine-tuning with Cross-Entropy (CE) does not align with ordinal evaluation such as Quadratic Weighted Kappa (QWK) (Yannakoudakis et al., 2011).

We propose a two-stage strategy for the Strict track: (i) initialize from BAREC-tuned AraBERT checkpoints, then (ii) fine-tune on the Strict splits with *Weighted Kappa Loss* (WKL), a differentiable surrogate aligned with QWK. We use specific input variants for each track, D3Tok for sentences and Word for documents, and adopt *max-level* aggregation for documents (label = hardest sentence) (Habash et al., 2025). This setup yields strong results at both levels.

2 Background

For education, ARA evaluates reading level to drive text selection, curriculum sequencing, and learner assessment (Vajjala, 2022). Early work relied on manually engineered features such as sentence length, word frequency, and syntactic complexity (Feng et al., 2010; Vajjala, 2022). While effective in controlled settings, such surface features often miss semantic and discourse cues, limiting robustness across genres and languages.

With large pretrained language models (PLMs) such as BERT (Devlin et al., 2019), the field shifted toward holistic fine-tuning with richer contextual representations; recent studies report strong gains for Transformer encoders in readability prediction (Martinc et al., 2021). We defer a focused survey of PLM approaches to Section 3 to avoid redundancy.

For Arabic, readability modeling is particularly challenging due to morphological richness,

Proceedings of The Third Arabic Natural Language Processing Conference, pages 274–279
November 8-9, 2025 ©2025 Association for Computational Linguistics

orthographic variation, and the coexistence of MSA with multiple dialects (Habash, 2010; Nassiri et al., 2023). Concurrent advances in Arabic PLMs—AraBERT (Antoun et al., 2020), ARBERT/MARBERT (Abdul-Mageed et al., 2021), and QARiB (Abdelali et al., 2021)—have delivered strong results across sentiment, dialect identification, and classification benchmarks (Abu Farha and Magdy, 2021); we discuss these in Related Work.

The BAREC resources standardize fine-grained Arabic readability: the shared task overview defines 19 ordinal levels and two evaluation settings ,General and Strict at the sentence and document levels (Elmadani et al., 2025b); the corpus paper details broad coverage for fine-grained labeling (Elmadani et al., 2025a); and the annotation guidelines specify procedures for consistent sentence level judgments (Habash et al., 2025). The **Strict** setting limits training to the official splits, and official evaluation uses *Quadratic Weighted Kappa* (QWK), motivating approaches that leverage pretrained encoders while aligning optimization with ordinal agreement.

3 Related Work

Early Arabic readability research adapted formulaic, feature-based methods from English, using shallow indicators (e.g., sentence length, word frequency, morphology) and classical ML; systems such as AARI and OSMAN established useful baselines but provide limited coverage of semantics and discourse and transfer poorly across domains (Al Tamimi et al., 2014; El-Haj and Rayson, 2016; Forsyth, 2014; Saddiki et al., 2018; Cavalli-Sforza et al., 2018). With pre-trained language models (PLMs) such as BERT (Devlin et al., 2019), richer contextual representations typically outperform feature-only models on readability prediction (Martinc et al., 2021; Lee et al., 2021). For Arabic NLP, AraBERT, ARBERT/MARBERT, and QARiB advance the state of the art across text classification tasks (Antoun et al., 2020; Abdul-Mageed et al., 2021; Abdelali et al., 2021), motivating PLM-based approaches to Arabic readability.

Readability labels are ordinal; however, optimizing nominal cross-entropy (CE) can misalign with QWK (Yannakoudakis et al., 2011; Martinc et al., 2021). Ordinal aware training includes (i) direct or surrogate optimization of QWK (e.g., WKL) (de la Torre et al., 2018), (ii) regression or threshold based ordinal classification, and (iii) pairwise or ranking objectives, which often reduce large magnitude errors relative to CE.

BAREC standardizes fine-grained Arabic readability with 19 levels at sentence and document scopes and defines Strict,it is data constrained track with settings using only official splits (Elmadani et al., 2025b,a; Habash et al., 2025). Official resources report PLM baselines and fine-grained evaluations; document labels follow the hardest sentence definition (Habash et al., 2025).

Complementary resources provide signals correlated with readability. The SAMER Readability Lexicon and SAMER Simplification Corpus supply leveled lexical cues and aligned simplification pairs, and recent work systematizes strategies for Arabic readability modeling (Al Khalil et al., 2020; Alhafni et al., 2024; Liberato et al., 2024). Orthographic or phonological indicators from large scale diacritized text enable features such as vowelization density and ambiguity reduction (Zaghouani et al., 2016).

Discourse signals arise from punctuation and boundary usage; Arabic punctuation annotation and a punctuated corpus support density of punctuation and restoration models (Zaghouani and Awad, 2016b,a). In learner contexts, correction annotated corpora provide error rate and edit operation statistics that proxy grammaticality and difficulty (Zaghouani et al., 2015). Word-level visualizations further illustrate fine-grained difficulty signals for assisted simplification (Hazim et al., 2022).

Within this landscape, our system starts from BAREC, then tunes PLMs, and continues training with a QWK aligned objective (WKL), targeting Strict track robustness and reduction of large ordinal errors.

4 System Overview

We participate in the **Sentence Strict** and **Document Strict** tracks of BAREC 2025, predicting fine-grained Arabic readability levels ($C{=}19$) under constrained training. The setting is challenging due to the large label space, skewed label distribution, and differences between sentence and document level detection.

4.1 Two-Stage Fine Tuning

We adopt a two-stage pipeline. **Stage 1 (warm start):** initialize from public AraBERT-based readability checkpoints released for BAREC (sentence: D3Tok input; document: Word input). These are trained with CE on BAREC and provide domain-driven representations (Antoun et al., 2020; Elmadani et al., 2025a). **Stage 2 (Strict specialization):** fine-tune only on the official Strict splits with WKL, a differentiable surrogate aligned with QWK, penalizing large ordinal errors more than small ones (de la Torre et al., 2018; Yannakoudakis et al., 2011).

Motivation: two-stage CE → WKL. The official metric for BAREC is *Quadratic Weighted Kappa* (QWK), which penalizes larger ordinal mistakes more. We therefore align optimization with evaluation by continuing training using a *Weighted Kappa Loss* (WKL). We use two stages instead of training WKL from scratch because: (i) A CE warm start from a BAREC-tuned checkpoint retains domain and split-specific signals, including tokenization and label priors over 19 levels. (ii) Direct WKL from an untuned PLM exhibited reduced stability on Strict (characterized by class imbalance and narrowly spaced labels), while CE produces a highly accurate classifier that WKL subsequently refines. (iii) Stage 2 emphasizes the mitigation of significant ordinal mistakes that influence QWK with minimal additional procedures. Specifically, upon CE convergence, we reload the checkpoint and transition to WKL with quadratic weights $w_{ij} = \left(\frac{i-j}{K-1}\right)^2$, K=19, lower the learning rate, and apply early stopping on dev QWK.

4.2 Model Architecture

Our model uses a Transformer encoder $\mathcal{E}$ (AraBERT family) with a linear head. Given input x, let $\mathbf{h}_{[\text{CLS}]} = E(x)_{[\text{CLS}]}$. The classifier computes

$$\boldsymbol{\ell} = W\,\mathbf{h}_{[\text{CLS}]} + \mathbf{b}, \quad \mathbf{p} = \text{softmax}(\boldsymbol{\ell}), \qquad (1)$$

where $W \in \mathbb{R}^{C\times d}$, $\mathbf{b} \in \mathbb{R}^{C}$, C=19, and d is the encoder hidden size. As shown in Equation 1, we map [CLS] to logits ℓ then to probabilities p.

4.3 Preprocessing and Optimization

We follow the shared-task input conventions for comparability: D3Tok for sentence-level inputs and Word for document-level inputs (matching the released checkpoints). No external data are used for Strict track. Hyperparameters, includeing learning rate, batch size and warmup, are tuned per track with early stopping on the Strict dev split.

4.4 Document Inference

Document labels are obtained via *max-level pooling* over sentence predictions (document level = level of the hardest sentence), consistent with the task definition (Habash et al., 2025).

4.5 Summary of Differences

In comparison to CE-only baselines using BAREC resources, our system (i) initiates from BAREC-optimized checkpoints, (ii) substitutes CE with WKL in stage 2 to synchronize training with QWK, and (iii) employs track-specific input variations (D3Tok vs Word) in accordance with the sentence/document configuration(Elmadani et al., 2025a).

5 Experimental Setup

We describe the datasets, input variants, model initialization, optimization, and evaluation protocol used in our Strict track sentence and document level experiments.

5.1 Data and Inputs

We use the BAREC 2025 resources, which provide sentence and document level readability annotations across 19 ordered levels (Elmadani et al., 2025b,a; Habash et al., 2025). We follow the official *Strict* splits and do not use external data. For the sentence track, inputs follow the **D3Tok** variant; for the document track, the **Word** variant, matching the released BAREC checkpoints.

5.2 Model Configurations

We adopt a two-stage strategy. **Stage 1** warm-starts from BAREC-tuned AraBERT checkpoints (sentence: D3Tok; document: Word) trained with CE (Antoun et al., 2020; Elmadani et al., 2025a). **Stage 2** specializes in the strict splits using WKL, a differentiable surrogate aligned with QWK, to better reflect ordinal evaluation.

5.3 Training Details

All runs use a single NVIDIA T4 (16 GB). We train with AdamW, initial learning rate 2×10^{-5}, batch size 16, linear decay with warmup ratio 0.1, and early stopping on dev QWK. Each model trains up to 10 epochs; the best dev QWK checkpoint is used for test submission.

5.4 Evaluation Metrics

The official metric is **QWK**,which quantifies agreement while punishing significant ordinal discrepancies. We provide QWK for validation, public test, and blind test partitions; accuracy is assessed only for diagnostic purposes.

Model (Tokenization)	Loss	Val QWK	Public Test QWK	Blind Test QWK
BAREC Official Baseline (Strict leaderboard)	–	–	–	**0.815**
Ours: AraBERTv2 (D3Tok)	WKL	**0.820**	**0.842**	**0.841**

Table 1: Sentence-level *Strict* results (QWK). Baseline taken from the official Strict-track leaderboard

Model (Tokenization)	Loss	Val QWK	Public Test QWK	Blind Test QWK
BAREC Official Baseline (Strict leaderboard)	–	–	–	**0.620**
Ours: AraBERTv2 (Word)	WKL	**0.820**	**0.828**	**0.790**

Table 2: Document-level *Strict* results (QWK). Baseline taken from the official Strict-track leaderboard

6 Results

We demonstrate strict track findings for sentence and document-level tasks, correlate them with corpus-paper baselines when relevant, and analyze observed mistake trends.

Metric. As stated in previous sections, we provide QWK using the official scorer in accordance with the BAREC procedure.

6.1 Sentence-Level (Strict Track)

Table 1 includes the official *Strict*-track baseline from the blind (final) leaderboard (QWK = 0.815). Our two-stage CE→WKL approach attains $0.842/0.841$ (public/blind) and improves over this baseline under the same Strict constraints.

6.2 Document-Level (Strict Track)

Table 2 reports our results alongside the official *Strict*-track baseline from the blind (final) leaderboard (QWK = 0.620). Our WKL specialization reaches $0.828/0.790$ (public/blind), showing gains on public test and a modest blind drop, suggesting sensitivity to domain shift and to max-level pooling.

Analysis. (1) The implementation of an ordinal-aware objective (WKL) aligns the training process with QWK and is consistent with trends observed in corpus papers, indicating that ordinal objectives demonstrate superior performance compared to CE on development datasets. (2) The sentence-level Strict scores obtained are $0.842/0.841$ for public and blind evaluations, respectively. These scores align with the general range of previous development split results reported on BAREC, even under more stringent training constraints. (3) Document-level blind performance (0.790) lags behind the public benchmark by approximately 0.04, suggesting a sensitivity to shifts in domain or topic as well as to max pooling techniques. Implementing hierarchical document encoders or utilizing calibrated/attention-based aggregation methods may enhance robustness further.
Reproducibility. We will release evaluation scripts, configs, and checkpoints upon acceptance.

7 Conclusion

We examined fine-grained Arabic readability in the *Strict* BAREC 2025 setting by initializing AraBERT from BAREC-tuned checkpoints and fine-tuning using a quadratic, ordinal-aware objective (WKL). An encoder utilizing track-specific inputs (D3Tok for sentences; Word for documents) and max-pooling for document label aggregation achieves **0.842/0.841** QWK (public/blind) at the sentence level and **0.828/0.790** at the document level. Errors are less frequent at higher magnitudes and tend to cluster between neighboring levels. Future research will focus on hierarchical document encoders, advanced aggregation methods beyond max-pooling, and efficient domain/task adaptation under strict constraints.

Limitations

This study is limited to the Strict track and uses only official data and BAREC-tuned checkpoints; generalization to other corpora, domains, or languages is untested. Document labels are obtained by max-pooling sentence predictions, which can be sensitive to outliers and intra-document variation. Compute constraints precluded extensive hyperparameter search or ensembling, and we report single-model runs. Finally, while we optimize an ordinal-aware loss and report QWK, broader evaluation (e.g., MAE, accuracy@±1) and statistical significance across multiple seeds, as well as genre/dialect–level error analysis, are left to future work.

Acknowledgments

This study was supported by the grant NPRP14C-0916-210015, awarded by the Qatar Research, Development and Innovation Council (QRDI).

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

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.

Ibrahim Abu Farha and Walid Magdy. 2021. Benchmarking transformer-based language models for Arabic sentiment and sarcasm detection. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 21–31, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Abdel Karim Al Tamimi, Manar Jaradat, Nuha Al-Jarrah, and Sahar Ghanem. 2014. Aari: automatic arabic readability index. *Int. Arab J. Inf. Technol.*, 11(4):370–378.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

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.

Violetta Cavalli-Sforza, Hind Saddiki, and Naoual Nassiri. 2018. Arabic readability research: Current state and future directions. *Procedia Computer Science*, 142:38–49. Arabic Computational Linguistics.

Jordi de la Torre, Domenec Puig, and Aida Valls. 2018. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognition Letters*, 105:144–154. Machine Learning and Applications in Artificial Intelligence.

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.

Mahmoud El-Haj and Paul Rayson. 2016. OSMAN — a novel Arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255, Portorož, Slovenia. European Language Resources Association (ELRA).

Khalid Elmadani, Nizar Habash, and Hanada Taha. 2025a. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025b. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Lijun Feng, Michael Elhadad, and Matt Huenerfauth. 2010. Automatic readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–62.

Jonathan Forsyth. 2014. Automatic readability prediction for modern standard arabic. Master's thesis, Brigham Young University, Provo, UT.

Nizar Habash. 2010. *Introduction to Arabic natural language processing*, 1 edition, volume 3 of *Synthesis Lectures on Human Language Technologies*. Morgan and Claypool Publishers.

Nizar Habash, Hanada Taha-Thomure, Khalid Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 47(1):141–179.

Naoual Nassiri, Violetta Cavalli-Sforza, and Abdelhak Lakhouaja. 2023. Approaches, methods, and resources for assessing the readability of arabic texts. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Hind Saddiki, Nizar Habash, Violetta Cavalli-Sforza, and Muhamed Al Khalil. 2018. Feature optimization for predicting readability of Arabic L1 and L2. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 20–29, Melbourne, Australia. Association for Computational Linguistics.

Sowmya Vajjala. 2022. Trends, limitations and open challenges in automatic readability assessment research. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5366–5377, Marseille, France. European Language Resources Association.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Wajdi Zaghouani and Dana Awad. 2016a. Building an arabic punctuated corpus. 2016(1):SSHAPP3148.

Wajdi Zaghouani and Dana Awad. 2016b. Toward an arabic punctuated corpus: Annotation guidelines and evaluation. In *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, volume 22.

Wajdi Zaghouani, Houda Bouamor, Abdelati Hawwari, Mona Diab, Ossama Obeid, Mahmoud Ghoneim, Sawsan Alqahtani, and Kemal Oflazer. 2016. Guidelines and framework for a large scale Arabic diacritized corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3637–3643, Portorož, Slovenia. European Language Resources Association (ELRA).

Wajdi Zaghouani, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Heider, and Kemal Oflazer. 2015. Correction annotation for non-native Arabic texts: Guidelines and corpus. In *Proceedings of the 9th Linguistic Annotation Workshop*, pages 129–139, Denver, Colorado, USA. Association for Computational Linguistics.

SATLab at BAREC Shared Task 2025: Optimizing a Language-Independent System for Fine-Grained Readability Assessment

Yves Bestgen
Statistical Analysis of Text Laboratory (SATLab)
Université catholique de Louvain
Place Cardinal Mercier, 10 1348 Louvain-la-Neuve, Belgium
yves.bestgen@uclouvain.be

Abstract

This paper presents SATLab's participation in the BAREC shared task on estimating the readability of sentences and documents. The proposed system is based on character n-grams fed into a support vector regression. A procedure is then applied to try to optimize the Quadratic Weighted Kappa, the main challenge measure, by tuning the decision thresholds used to transform continuous values into ordered categories. Performance is significantly lower than that of the best systems, but nevertheless superior to that of several deep learning approaches.

1 Introduction

Being able to estimate the level of difficulty of a sentence, paragraph, or text has long been an important goal in education (Dale and Chall, 1948). It has been repeatedly demonstrated that students learn better when the texts they are asked to understand are neither too simple nor too complex for them (Vajjala, 2022). It is also important in our society that documents produced by administrations, journalists, and even generative AI can be understood by their recipients while remaining sufficiently interesting to read. Striking the right balance between uninteresting simplicity and discouraging complexity requires the ability to accurately assess readability[1].

Conducting research in this field requires corpora annotated by experts according to readability level. As highlighted in Vajjala and Lučić (2018), the vast majority of available datasets are composed of texts. This is certainly a relevant level of granularity, but a text that is generally simple may contain very complex sentences, which are well beyond the comprehension of the average reader. Corpora in which sentences have been annotated according to their readability level are very rare, and even more so in languages other than English (Hazim et al., 2022; Liberato et al., 2024; Vajjala and Lučić, 2018). Very recently, Elmadani et al. (2025b) developed a corpus for Arabic: the Balanced Arabic Readability Evaluation Corpus (BAREC), which contains 69,441 sentences classified into 19 readability levels. This corpus is at the heart of the BAREC Shared Task 2025 (Elmadani et al., 2025a), which invites participants to develop an automatic approach to estimating the readability of this material. This paper presents SATLab's participation in this shared task.

As in many areas of NLP, deep learning approaches and the use of pre-computed embeddings have proven to be the most effective for estimating the readability of documents or sentences (Lee et al., 2021; Naous et al., 2024; Martinc et al., 2021). However, Vajjala and Lučić (2018) achieved excellent results with a much simpler system, using character n-grams, a well-established approach in computational linguistics (Damashek, 1995), which are fed into some classical supervised approaches such as logistic regression. The advantage of such an approach is that it is completely language-independent, but also that it is does not requires additional resources. For a number of years, SATLab has specialized in using this type of approach to solve complex tasks such as predicting eye saccades during reading (Bestgen, 2021a) or identifying offensive content and hate speech in languages with few linguistic resources (Bestgen, 2021b). Using such a language-independent system in the BAREC task will allow for at least a partial evaluation of the benefits provided by complementary knowledge such as pre-computed embeddings and by the use of far more complex architectures. However, it should be noted that the experiments in Vajjala and Lučić (2018) were conducted on less than 200 texts obtained by asking teachers to rewrite English newspaper articles at

[1] It also requires the ability to assess the reader's language proficiency, but this is an issue that will not be addressed here (Bestgen, 2017)

three levels of ESL learners (elementary, intermediate, and advanced). The criterion was therefore to distinguish between these three levels of complexity. It is far from obvious that character n-grams will be equally effective in accurately assessing the fine-grained readability level of Arabic sentences, as is the case in the BAREC 2025 task described below.

2 The BAREC Shared Task 2025

The BAREC Shared Task 2025 (Elmadani et al., 2025a) is based on the Balanced Arabic Readability Evaluation Corpus (BAREC), which contains 1,922 Arabic documents whose sentences (N = 69,441) have been evaluated by annotators in terms of readability on a 19-point scale, 19 indicating the most difficult sentences to understand (Elmadani et al., 2025b; Habash et al., 2025). The corpus covers many genres and topics intended for different target audiences. Annotated examples from the corpus are presented in the two papers mentioned above.

The goal is to develop an automatic model capable of estimating readability levels. These estimates can be made at the sentence or document level. Since the readability of the documents was not directly annotated, the organizers decided that it was equal to the readability level of its most difficult sentence. The material provided by the organizers for the development of the system consists of the entire BAREC corpus. It is divided into three subcorpora: the Learning subcorpus (L) consisting of 1,518 documents and 54,845 sentences, the Development subcorpus (D) consisting of 194 documents and 7,310 sentences, and the Public Test subcorpus (PT) consisting of 210 documents and 7,286 sentences.

Three tracks are available to participants. For the first track, known as “strict,” the only readability annotated data that can be used are those from BAREC corpus. For the second track, participants can also use the training set of SAMER Corpus and the SAMER Lexicon (Alhafni et al., 2024; Al Khalil et al., 2020), while for the third track, any publicly available resource can be used. SATLab participated in both tasks of the “strict” track.

The main metric for the challenge is the Quadratic Weighted Kappa (QWK). Several other metrics were also proposed by the organizers, such as accuracy, the percentage of cases where reference and prediction classes match in the 19-level scheme. These will not be discussed here because, as pointed out by Elmadani et al. (2025b), different approaches are needed to optimize a system for these different metrics. The SATLab system will therefore be optimized for the main metric, QWK.

The baseline proposed by the organizers is described in Elmadani et al. (2025b). It is a highly effective baseline which uses, among other things, fine-tuning the very effective Arabic BERT-based models.

3 System Overview

The system proposed by SATLab for the Sentence-level task is mainly based on the character n-grams of the sentences to be analyzed. Some statistics about the sentences, such as their length in characters, and some variables provided in the corpus, such as the annotator, are also taken into account. All these indices are fed into a very classic supervised learning procedure, support vector regression (SVR). A regression-type approach was chosen because Elmadani et al. (2025b) showed this kind of approaches were particularly effective when the metric was QWK.

SVR produces a continuous value that must be converted to integers in the 19-ordinal category system used for readability annotation. This can be done in a very simple way, by rounding these continuous values to the nearest integer and ensuring that none of the values obtained are less than 1 or greater than 19. However, Beckham and Pal (2017) showed that it was possible to improve the QWK of a predictor by modifying the loss function. Their approach is relatively complex, at least for me. For this reason, a simple procedure was developed to try to optimize the QWK by tuning the decision thresholds used to transform continuous values into ordered categories.

The system developed for the document-level task is also based on a SVR mainly fed with the continuous readability estimates from the sentence-level system described above. The features used include the lowest readability value (highest score on Readability Level 19) returned by the Sentence-level system for the document, a series of features encoding the proportion of sentences in the document that have a predicted value equal to a given value (after rounding), and a few global statistics and variables provided in the corpus. The SVR continuous readability estimates were converted to the 19-ordinal category system by the procedure used for the Sentence-level track.

4 Implementation

Almost all of the analyses were performed using a series of custom SAS programs. The QWK optimization was programmed in C. The supervised learning procedure used is the LibLinear L2-regularized L2-loss SVR dual (Fan et al., 2008).

Both systems were optimized on the combination of L and D sets using a 9-fold cross-validation procedure (CV9). These folds were stratified by document, with all sentences from a given document placed in the same fold. In order to ensure that folds contained similar numbers of sentences, the random distribution also took into account the length of the documents in terms of sentences. This CV9 step led to the following parameters being set for the Sentence-level track.

- Character n-grams with n in [1, 6], which occurs at least 10 times in the dataset. These features were weighted by the Sublinear Tf-Idf and then L2-normalised.
- Two global statistics: the log-transformed number of characters and number of different characters.
- Four one-hot encoded variables provided in the corpus: Annotator, Source, Domain and Text Class.
- The SVR regularization parameter and bias were set to 3 and 0.1, respectively.

As explained above, the SVR continuous readability estimates were converted to integers in the 19-ordinal category system by a handcrafted function that attempts to optimize the QWK. The OptiK function takes the SVR continuous readability estimates as input. The thresholds (T) for rounding are initially set to the usual values for rounding to an integer and the QWK is calculated. This value is provisionally considered to be the maximum QWK. Next, the procedure randomly chooses a threshold (Ti) and searches between Ti-1 and Ti+1 for the value for that threshold that produces the largest QWK, starting in the middle of the range of values to be tested and advancing in each direction in turn. This procedure may seem insignificant. However, it favors values in the middle of the interval when there are multiple occurrences of the maximum value. If this maximum value is greater than the current maximum QWK, it replaces it, and Ti is set to the new threshold. This procedure is repeated 150 times, an arbitrary number chosen after some trial and error.

	L->D	L->PT	L+D->PT
No OptiK	76.9	78.1	78.7
OptiK	78.2	79.6	80.5

Table 1: QWK for the Sentence-level

It is not advisable to apply the OptiK function on the data that has been used to train the predictive model, due to model overfitting for this data. It is therefore preferable to use predicted data. Two scenarios were used:

- Train the predictive model on the L set and apply it to the D and PT sets. Then, 1) optimize the thresholds on the D set and evaluate them on the PT set, and 2) optimize the thresholds on the PT set and evaluate them on the D set.
- Train the predictive model on the combination of the L and D sets in CV9, combine the predictions for the 9 folds into a single dataset, optimize the thresholds on it, and evaluate them on the PT set.

For the Document-level task, the predictive model was built based on the following features and parameters:

- The lowest readability value (highest score on Readability Level 19) returned by the Sentence-level system for the document.
- Twelve features encoding the proportion of sentences in the document that have a predicted integer round score equal to a given value from 8 to 19.
- Two global statistics: the log-transformed number of sentences and number of words in the document.
- Three one-hot encoded variables provided in the corpus: Source, Domain and Text Class.
- The SVR regularization parameter and bias were set to 6 and 0.5, respectively.

The SVR continuous readability estimates were converted to the 19-ordinal category system using the OptiK procedure described above.

	L+D (CV9)			L+D->PT
	Mean	Min	Max	
No OptiK	72.2	67.1	77.5	64.3
OptiK				67.1

Table 2: QWK for the Document-level

	Sentence		Document	
	Final	No OptiK	Final	No OptiK
Best	87.5		87.4	
SATLab	82.3	80.2	77.6	73.3
Baseline	81.5		62.0	

Table 3: QWK for the BT set

5 Results

This section presents the performance of the proposed system, first on the D and PT sets, and then on the real challenge, i.e., the Blind Test set (BT). The latter consists of 100 documents and 3,420 sentences.

5.1 Public evaluation sets

Table 1 presents the QWK for the different public evaluation sets for the Sentence-level task. We can see that the PT set is a little simpler than the D Set and that adding the D to the L set for learning improves performance, which is obviously to be expected. Above all, we observe that optimizing the QWK brings a benefit of 1.3% and 1.8% in QWK, which does not seem negligible. The best performance obtained on the PT set is slightly higher than that obtained by the Baseline system (QWK = 80.2). Exceeding this value was one of SATLab's objectives, since the Baseline system uses, among other things, fine-tuning the very effective Arabic BERT-based models (Elmadani et al., 2025b).

The material for the document-level task is relatively small for supervised learning procedures. For this reason, the conditions evaluated are different from those used for the sentence-level task. Learning was performed on the combination of the L and D sets in CV9, QWK optimization on the 9 predicted folds, and final evaluation on the PT set.

The QWKs are significantly weaker for this task (Table 2). This is likely due to the inaccuracy of the Sentence-level model, which produces an overly imperfect estimate of the readability level of the most difficult sentence in a document. It is particularly noteworthy that CV9 performance varies greatly depending on the fold. There is therefore a high degree of instability in the results, probably due to the relatively small number of documents in each fold (N = 190) and in the PT set (N = 210).

5.2 Challenge results: BT set

The main question that this study attempts to answer is that of the performance level of a system based on indices as simple and as unspecific to the task as character n-grams compared to much more complex systems, such as those using precomputed embeddings. As reference points, I chose the Baseline system, described in Elmadani et al. (2025b), and the top-ranked system, !MSA, assuming that it also uses sophisticated techniques. To analyze this BT set, the complete public material (L+D+PT) was used for learning.

Table 3 shows that the SATLab system is capable of outperforming systems that use fine-tuning of BERT-based models, but that QWK optimization is essential to achieve this result. The difference with the best system is clearly significant (5.2%) and justifies the use of more complex models than an SVR on character n-grams, as proposed by SATLab.

It should be noted that the QWK of the Baseline system for documents is significantly lower than the QWK of all other systems that participated in this task. It seems likely that this system's prediction for a document is simply equal to the highest predicted score for the sentences in that document, without taking into account any other features or new learning. There is no doubt that a higher performance could have been achieved.

5.3 Impact of OptiK on thresholds

The results presented above indicate that QWK optimization is essential for the system to achieve a competitive score. This trick, if not used by other systems, somewhat distorts the comparison with them. Indeed, it is reasonable to assume that they could have improved their QWK in this way.

In order to gain a clearer understanding of the effects of OptiK on the thresholds used to transform SVR scores into categories, Figure 1 shows the thresholds obtained by this procedure for submission to the Sentence-level task. The bottom line shows the range of predicted values from the SVR. The middle line simply indicates the thresholds usually used when rounding a real number to a whole number. The top line shows the thresholds obtained using the OptiK procedure. As can be seen, some thresholds are significantly modified. For example, the range of values corresponding to category 1

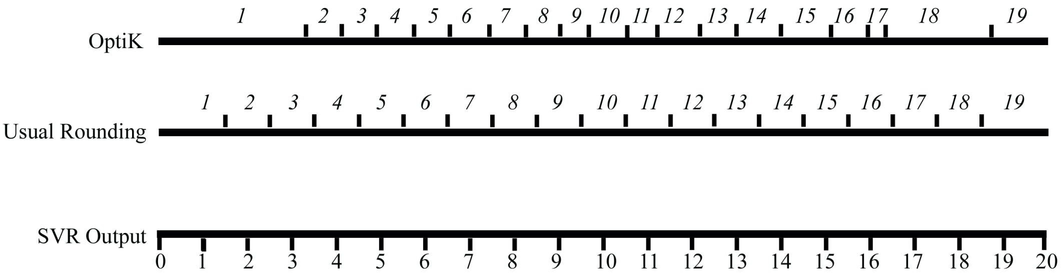


Figure 1: Effects of OptiK on the thresholds

is greatly expanded, while that corresponding to category 17 is smaller than what is obtained using a standard rounding procedure. We can even see that the continuous values corresponding to several categories do not cover the usual range of values (see categories 2 to 6, 9, 10, 17).

6 Conclusion

This paper presents SATLab's participation in the BAREC shared task. The proposed system relies almost exclusively on character n-grams, which are used by an SVR to estimate the readability of Arabic sentences. A post-processing procedure is then applied to the predicted values to optimize the main measure of the challenge: QWK. This system ranks 16th out of 24 in the Sentence-level task when all participating teams are taken into account, and 13th out of 16 in the official ranking composed of participating teams that have published a report about their system. It is 4th out of 8 in the official Document-level ranking, each time for the Strict track. These performances make it more effective than systems using precomputed embeddings, but it is important to remember that a significant part of its effectiveness comes from the QWK optimization procedure and that it is likely that several other systems did not use such a trick.

As for the shared task itself, I think it could be interesting to reevaluate the document-level task. In particular, the analyses conducted in CV9 showed significant variability in performance depending on the fold. The comparison of QWKs on the PT set (SATLab = 67.1) and BT sets (SATLab = 73.3) confirms this significant variability. It could be related to the small size of these samples, which means that changing a few predictions can significantly affect the QWK. It might also be interesting to replace the current procedure for determining the readability level of a document (that of the most difficult sentence) with an annotation made by experts.

Acknowledgments

The author wishes to thank the organizers of this shared task for putting together this valuable event and the reviewers for their very constructive comments. He is a Research Associate of the Fonds de la Recherche Scientifique - FNRS (Fédération Wallonie Bruxelles de Belgique).

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING, 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Christopher Beckham and Christopher Pal. 2017. A simple squared-error reformulation for ordinal classification. *Preprint*, arXiv:1612.00775.

Yves Bestgen. 2017. Beyond single-word measures: L2 writing assessment, lexical richness and formulaic competence. *System*, 69:65–78.

Yves Bestgen. 2021a. LAST at CMCL 2021 shared task: Predicting gaze data during reading with a gradient boosting decision tree approach. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 90–96, Online. Association for Computational Linguistics.

Yves Bestgen. 2021b. A simple language-agnostic yet strong baseline system for hate speech and offensive content identification. In *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation*, CEUR Workshop Proceedings. CEUR-WS.org.

Edgar Dale and Jeanne Chall. 1948. A formula for predicting readability: Instructions. *Educational Research Bulletin*, pages 37–54.

Marc Damashek. 1995. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843–848.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, Vienna, Austria. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 47(1):141–179.

Tarek Naous, Michael J Ryan, Anton Lavrouk, Mohit Chandra, and Wei Xu. 2024. ReadMe++: Benchmarking multilingual language models for multi-domain readability assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12230–12266, Miami, Florida, USA. Association for Computational Linguistics.

Sowmya Vajjala. 2022. Trends, limitations and open challenges in automatic readability assessment research. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5366–5377, Marseille, France. European Language Resources Association.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

MorphoArabia at BAREC Shared Task 2025: A Hybrid Architecture with Morphological Analysis for Arabic Readability Assessment

Fatimah Emad Eldin
Department of Computer and Information Sciences
Faculty of Graduate Studies for Statistical Research, Cairo University
12422024441586@pg.cu.edu.eg

Abstract

This paper presents MorphoArabia, a system developed for the BAREC Shared Task 2025 on Arabic Readability Assessment. The approach is centered on the hypothesis that deep morphological analysis is fundamental for modeling the complexity of the Arabic language. A regression model was fine-tuned on AraBERTv2 with morphologically-aware tokenization via CAMeL Tools. Various configurations were explored for the strict, constrained, and open tracks, including a hybrid model with seven engineered lexical features. The system demonstrated highly competitive performance, securing top-10 rankings in all six subtasks and achieving a peak Quadratic Weighted Kappa (QWK) of 84.2% on the strict sentence-level task. All code and models are publicly available to facilitate future research.

1 Introduction

Automatic Readability Assessment for Arabic is a challenging task, primarily due to the language's rich and complex morphology (Liberato et al., 2024). Consequently, traditional readability formulas that rely on surface-level features are often insufficient for capturing the nuanced difficulty of Arabic text (Al-Tamimi et al., 2014). The BAREC Shared Task 2025 (Elmadani et al., 2025a) addresses this by providing a large-scale, fine-grained dataset annotated on a 19-level readability scale. This paper introduces MorphoArabia, a system designed to address this challenge by explicitly modeling Arabic morphology. The core hypothesis is that a model's performance can be significantly improved by providing it with text analyzed at the morpheme level. This hypothesis is tested across the three competition tracks:

- **Strict Track:** A fine-tuned AraBERTv2 (Antoun et al., 2020) regression model using only the official BAREC corpus.
- **Constrained Track:** A hybrid architecture augmenting the base model with seven engineered lexical features derived from the SAMER corpus (Alhafni et al., 2024).
- **Open Track:** The base regression model trained on a combination of the BAREC and DARES corpora (El-Haj et al., 2024).

The system achieved competitive results across all tracks, notably securing 2nd place in both the strict and open document-level tasks, validating the effectiveness of the morphologically-aware approach. Key findings include the superior performance of regression over classification for this task, along with the challenges of harmonizing datasets with disparate annotation scales. To ensure reproducibility, all code and models are available on GitHub[1] and Hugging Face[2].

2 Background and Related Work

2.1 Task Description

The BAREC Shared Task 2025 utilizes the Balanced Arabic Readability Evaluation Corpus (BAREC), a dataset exceeding 1 million words annotated for readability assessment (Elmadani et al., 2025b). The task's 19-level annotation scheme is detailed in the official guidelines (Habash et al., 2025). The task comprises two primary goals:

- **Task 1: Sentence-level Readability Assessment:** Predict a readability score (1-19) for a given Arabic sentence.
- **Task 2: Document-level Readability Assessment:** Predict an overall document readability score, defined by the highest score of any sentence within it.

[1] https://github.com/astral-fate/barec-Arabic-Readability-Assessment

[2] https://huggingface.co/collections/FatimahEmadEldin/barec-shared-task-2025-689195853f581b9a60f9bd6c

Proceedings of The Third Arabic Natural Language Processing Conference, pages 286–296
November 8-9, 2025 ©2025 Association for Computational Linguistics

Participation was offered across three tracks: Strict, Constrained, and Open, each with distinct data constraints.

2.2 Related Work and Available Datasets

The landscape of Arabic NLP resources is extensively documented by initiatives like **Masader** (Alyafeai et al., 2021; Altaher et al., 2022). For readability assessment, several key resources include:

- **DARES** (El-Haj et al., 2024): A corpus of school textbooks with fine-grained (G1-G12) and coarse-grained labels.
- **OSMAN** (El-Haj and Rayson, 2016): A readability metric providing a continuous 0-100 score.
- **ARC-WMI** (AL-Dayel et al., 2018): A medical corpus with three difficulty levels.
- **SAMER Project**: This project introduced a lexicon with a 5-level scale (L1-L5) (Elmadani et al., 2025b). A related Google Docs add-on was also developed for word-level readability visualization (Hazim et al., 2022).
- **SAMER Corpus** (Alhafni et al., 2024): A text simplification corpus with parallel texts across multiple readability levels, used for comprehensive modeling approaches ranging from rule-based methods to pretrained language models (Liberato et al., 2024).

3 System Overview

The system employs two main architectures: a base regression model for the Strict and Open tracks, and a hybrid model for the Constrained track, which incorporates engineered features.

3.1 Morphological Analysis

The preprocessing pipeline utilized the CAMeL Tools d3tok analyzer (Obeid et al., 2020) for external datasets such as SAMER and DARES. This tool performs deep morphological analysis by disambiguating words in context and segmenting them into constituent morphemes, capturing complexities often missed by standard tokenization.

3.2 Feature Engineering

For the Constrained track, the system was enhanced with a hybrid architecture integrating engineered lexical features with the Transformer model's contextual understanding. Seven numerical features were engineered for each sentence using the SAMER lexicon to provide explicit signals about text complexity. A detailed description of these features is provided in Table 3 in Appendix A. The final sentence representation is created by concatenating the Transformer's '[CLS]' token embedding with this 7-dimensional feature vector, which is then passed to a regression head for prediction.

3.3 Level Mapping for External Datasets

To augment training data for the Constrained and Open tracks, external corpora were incorporated, necessitating mapping their distinct annotation scales to the 19-level BAREC scale.

- **DARES Corpus**: For the Open track, ”G1-G12” labels were directly mapped to BAREC levels 1-12.

- **SAMER Corpus**: For the Constrained track, SAMER's 5-level scale was harmonized with BAREC's 19-level scale. A heuristic mapped SAMER levels L3, L4, and L5 to BAREC values 4, 10, and 16, respectively.

This heuristic mapping process was identified as a potential source of noise and variance, potentially impacting model performance by introducing inconsistencies.

4 Experimental Setup

4.1 Datasets

Datasets and distributions were defined by each competition track. All data was preprocessed into the d3tok format before training.

- **Strict Track**: Limited to the official **BAREC** corpus.

  - **Sentence-level**: BAREC training (54,845 sentences) and development (7,310 sentences) and (7,286) test records, for the development phase, and (3,417) for the blind testing phase.
  - **Document-level**: Official document splits for the development phase is: 1,518 training, 194 development, 210 testing, and (100) for the tblind esting phase.

- **Constrained Track**: **BAREC** training data augmented with the **SAMER Corpus**.

  - Combined **sentence-level training set**: 97,874 sentences.

Track	Task	Dev (QWK)	Public Test (QWK)	Blind Test (QWK)	Hugging Face Model
Strict	Sentence	82.64	83.61	**84.2**	[Link]
	Document	71.07	65.91	79.90	[Link]
Constrained	Sentence	80.07	80.71	82.9	[Link]
	Document	75.60	62.70	75.5	[Link]
Open	Sentence	83.10	82.06	83.9	[Link]
	Document	72.85	57.11	**79.2**	[Link]

Table 1: Final QWK scores and Hugging Face models for each task. For document-level tasks, scores were derived from the sentence-level model by assigning each document the highest readability score found among its sentences.

  - Original BAREC development set (7,310 sentences) used for validation.

- **Open Track**: This track permitted the use of external data, with experiments primarily focusing on combining the **BAREC** and **DARES** datasets. Different data configurations were explored to optimize performance for both sentence-level and document-level tasks. More details on the data distributions for the Open track can be found in Appendix B.

4.2 Training and Hyperparameters

Models were fine-tuned with varied hyperparameters, primarily adjusting learning rate (2e-5-5e-5) and epochs (6-20). All models used the AdamW optimizer and an early stopping callback monitoring validation QWK score. A detailed summary of the hyperparameter values for the best performing models can be found in Appendix E.

4.3 Evaluation Metrics

The primary metric for evaluation is the Quadratic Weighted Kappa (QWK) (Cohen, 1968), as defined in Equation 1.

$$\kappa_w = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}} \qquad (1)$$

In this formula, O is the matrix of observed agreement, E is the matrix of expected agreement, and $w_{ij} = (i - j)^2$ is the quadratic weight matrix that penalizes larger disagreements more severely.

5 Results

The system demonstrated strong and consistent performance across all competition tracks. As summarized in Table 1, the top performance achieved was a sentence-level Quadratic Weighted Kappa (QWK) of 84.2% in the Strict track and a document-level QWK of 79.2% in the Open track. Full configurations for the best-performing models are detailed in Appendix E (Table 7).

5.1 Sentence-Level Analysis

The system achieved a highly competitive QWK score of 84.2% on the Strict track, earning a 7th-place rank on the official leaderboard. This performance is nearly identical to the official BAREC benchmark score of 84.4% (Elmadani et al., 2025b). The minor difference is attributed to variations from the custom morphological analysis used in this work, as opposed to the official pre-processed dataset provided by the organizers.

In the Constrained track, the hybrid model yielded a QWK of 82.9%, which earned a 3rd-place ranking. For the Open track, a QWK of 83.9% was attained by augmenting the training data with the DARES corpus, resulting in a 2nd-place ranking. It was noted, however, that neither of these results exceeded the performance observed in the Strict track. This observation reinforces the notion that the difficulties inherent in mapping different annotation scales can introduce label and domain variance, which may temper the performance improvements expected from additional data.

5.2 Document-Level Analysis

For the document-level task, no models were directly fine-tuned on full documents. Instead, the assessment was derived from the corresponding sentence-level models by assigning each document the maximum readability score predicted among all its sentences. A substantial increase in the QWK was observed between the development phase and the final blind test evaluation. The document-level QWK for the Strict track increased from a development score of 62.37% to a final blind test score of 79.9%, achieving a 2nd-place rank in the official evaluation (Tables 1, 7). A 3rd-place ranking was secured in the Constrained track with a score of 75.5%. In the Open track, the score increased from a development QWK of 60.48% to 79.2%, which also ranked 2nd (Tables 1, 7). This considerable improvement suggests the blind test set featured a different distribution of document complexity, one where difficulty was determined by a few outlier sentences.

5.3 Comparison with Official Baseline

A direct comparison with the final baseline results released by the shared task organizers reveals that the MorphoArabia system demonstrated a significant performance improvement across all six subtasks. The official baseline scores are sourced from the final leaderboards published on the BAREC Shared Task website.

‘‘‘

As shown in Table 2, MorphoArabia outperformed the baseline by a notable margin in every category. The most substantial gains were observed in the document-level tasks, where the system's max-score aggregation strategy proved highly effective, leading to improvements of +17.9, +13.5, and +17.2 QWK points for the Strict, Constrained, and Open tracks, respectively. The sentence-level tasks also showed consistent improvements, confirming the robustness of the morphologically-aware approach.

5.4 Hyperparameter and Data Ablation Analysis

The optimal configuration for the sentence-level task did not yield the best performance for the document-level task. The model from Experiment 2 achieved the highest sentence-level QWK on the blind test set (83.9%), whereas the model from Experiment 5 yielded the top document-level score (79.2%). Notably, the best sentence-level performance on the validation set (83.6%) was achieved in Experiments 3 and 4, not Experiment 2 (Table 6). This suggests that the document-level task, being highly sensitive to single-sentence errors, benefits from a validation set that better mirrors the complexity distribution of the augmented training data. Furthermore, experiments combining all three datasets (BAREC, SAMER, and DARES) did not lead to superior results, highlighting that more data is not always beneficial when significant label and domain variance is introduced, as detailed in Appendix C.2 (Table 6).

5.5 Ablation on Task Formulation

To validate the problem formulation, an ablation study was conducted comparing the primary regression approach (predicting a continuous score) against a multi-class classification alternative (predicting one of 19 discrete levels). For this comparison, the classification models were tested using a custom, non-morphological data normalization pipeline in place of the d3tok Morphological Analyzer (see Appendix C.1). The classification approach consistently yielded inferior performance compared to the morphologically-aware regression model, as detailed in Appendix C.2 (Table 5). This result confirmed that the regression framework was the more effective formulation for this task.

5.6 Morphological Error Analysis

A key source of error was identified as preprocessing artifacts. Appendix F (Table 8) provides examples where the d3tok analyzer failed to produce a morphological analysis, instead inserting a NOAN (No Analysis) token. This occurs for words not in its vocabulary or for words with valid but less frequent morphological forms. This noise, introduced during data augmentation, can degrade model reliability, especially for the document-level task.

6 Discussion

The performance of the MorphoArabia system, summarized in Table 1, validates the core hypothesis that a morphologically-aware model is highly effective for Arabic readability assessment. The top score achieved in the Strict sentence-level track (84.2% QWK) was highly competitive, nearly matching the official BAREC benchmark of 84.4% and underscoring the success of this foundational approach.

A key observation from the results is that models augmented with external data (Constrained and Open tracks) did not surpass the baseline model trained exclusively on the BAREC corpus. This suggests that the benefits of additional data were negated by noise introduced when harmonizing disparate datasets. Heuristically mapping different annotation scales (e.g., SAMER's 5-level and DARES's 12-level) to BAREC's 19-level schema likely introduced significant label and domain variance, highlighting that annotation quality and consistency are paramount.

Track	Task	MorphoArabia (QWK)	Official Baseline (QWK)
Strict	Sentence	**84.2%**	81.5%
	Document	**79.9%**	62.0%
Constrained	Sentence	**82.9%**	81.5%
	Document	**75.5%**	62.0%
Open	Sentence	**83.9%**	81.5%
	Document	**79.2%**	62.0%

Table 2: Comparison of final blind test QWK scores between MorphoArabia and the official shared task baseline.

The document-level assessment strategy, which assigned the maximum sentence score to the document, proved effective, securing second-place rankings in two tracks. The significant QWK score increase in the blind test suggests its distribution contained documents whose difficulty was driven by a few outlier sentences, a characteristic well-suited to the chosen max-score approach. Additionally, ablation studies confirmed that formulating the task as regression consistently outperformed a multi-class classification approach (Appendix C.2, Table 5).

Despite the system's success, two primary limitations were identified. First, reliance on the d3tok analyzer made the system susceptible to preprocessing artifacts. It failed to parse out-of-vocabulary or infrequent words, inserting a NOAN (No Analysis) token (see Appendix F). This introduced noise by depriving the model of crucial morphological information, a particularly detrimental issue for the document-level task where a single unanalyzed word can determine the overall score. Second, the simplistic, direct mapping used for data augmentation presents a significant challenge, as it fails to account for subtle differences in annotation criteria between corpora, leading to label noise.

7 Conclusion

This paper presented MorphoArabia, a system developed for the BAREC Shared Task 2025 on Arabic readability assessment. The system was centered on the hypothesis that a deep morphological approach is fundamental to modeling the nuances of Arabic text complexity. It employed two main architectures: a fine-tuned AraBERTV2 regression model and a hybrid model enhanced with seven engineered lexical features for the Constrained track. The results demonstrated highly competitive performance, securing 2nd place in the strict and open document-level tasks and achieving a peak QWK of 84.2% on the strict sentence-level task, thereby validating the effectiveness of the core approach. Despite its success, this work identified two primary limitations.

First, the process of harmonizing external datasets (SAMER and DARES) with different annotation scales introduced label noise, which may have tempered performance gains on the augmented tracks. Second, the system's reliance on the d3tok analyzer made it susceptible to preprocessing artifacts, where out-of-vocabulary or morphologically infrequent words were not analyzed, potentially degrading model reliability.

These limitations inform several directions for future work. Research could focus on more sophisticated domain adaptation techniques to better integrate external corpora and mitigate the effects of label variance. Another avenue is to improve the robustness of the morphological preprocessing pipeline, either by fine-tuning the analyzer on a broader vocabulary or by developing strategies to handle analysis failures.

Finally, exploring architectures that are directly fine-tuned on the document-level task, rather than relying on sentence-level aggregation, presents a promising path toward further performance improvements. In the interest of open science and to facilitate future research, all code for preprocessing, training, and evaluation, alongside the final fine-tuned models for each track, have been made publicly available. The experimental code is accessible on GitHub, and the models are hosted on the Hugging Face Hub, providing a strong and reproducible baseline for future work in Arabic readability assessment.

Acknowledgments

Acknowledgment is made to the organizers of the BAREC Shared Task 2025, the developers of CAMeL Tools, and the creators of the DARES and SAMER corpora for their invaluable resources.

References

Abeer AL-Dayel, Hend Al-Khalifa, Sinaa Alaqeel, Norah Abanmy, Maha Al-Yahya, and Mona Diab. 2018. ARC-WMI: Towards building Arabic readability corpus for written medicine information. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

Abdel-Karim Al-Tamimi, Manar Jaradat, Nuha Aljarrah, and Sahar Ghanim. 2014. AARI: Automatic Arabic Readability Index. *The International Arab Journal of Information Technology*, 11(4):384–391.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Yousef Altaher, Ali Fadel, Mazen Alotaibi, Mazen Alyazidi, Mishari Al-Mutairi, Mutlaq Aldhbuiub, Abdulrahman Mosaibah, Abdelrahman Rezk, Abdulrazzaq Alhendi, Mazen Abo Shal, Emad A. Alghamdi, Maged S. Alshaibani, Jezia Zakraoui, Wafaa Mohammed, Kamel Gaanoun, Khalid N. Elmadani, Mustafa Ghaleb, Nouamane Tazi, Raed Alharbi, and 2 others. 2022. Masader plus: A new interface for exploring +500 arabic nlp datasets. *Preprint*, arXiv:2208.00932.

Zaid Alyafeai, Maraim Masoud, Mustafa Ghaleb, and Maged S. Al-shaibani. 2021. Masader: Metadata sourcing for arabic text and speech data resources. *Preprint*, arXiv:2110.06744.

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 Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

Mahmoud El-Haj and Paul Rayson. 2016. OSMAN — a novel Arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255, Portorož, Slovenia. European Language Resources Association (ELRA).

Mo El-Haj, Sultan Almujaiwel, Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2024. DARES: Dataset for Arabic readability estimation of school materials. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 103–113, Torino, Italia. ELRA and ICCL.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. 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.

A Engineered Lexical Features

This appendix provides a detailed description of the seven engineered lexical features that were integrated into the hybrid model for the Constrained track. These features were designed to provide the model with explicit, interpretable signals about text complexity, complementing the deep contextual understanding from the Transformer architecture. Table 3 lists these features.

B Data Distribution for the Open Track

This appendix outlines the specific data configurations used for the Open track experiments. Since this track allowed for external data, different strategies were employed to combine the BAREC and

No.	Description	No.	Description
1	Sentence length (total characters)	5	Maximum word difficulty in the sentence
2	Number of words in the sentence	6	Count of ’hard’ words (difficulty score > 4)
3	Average word length in characters	7	Fraction of Out-of-Vocabulary (OOV) words
4	Mean word difficulty (from lexicon)		

Table 3: Description of the seven engineered lexical features used in the hybrid model.

DARES datasets to optimize performance for both the sentence-level and document-level tasks.

- **Sentence-Level Task Data Configuration**:
  - **Training Set**: A combination of the BAREC dataset and the official train/development splits of the DARES dataset, totaling 64,548 records. This consisted of:
    - 9,703 records from the DARES training set.
    - 1,380 records from the DARES development set.
    - Remaining records from the BAREC dataset.
  - **Validation Set**: 8,690 records.
- **Document-Level Task Data Configuration**: A more specific data splitting strategy was employed.
  - **Training Set**: Consisted of the entire BAREC dataset combined with 85% of the complete DARES dataset (merging its train, development, and test splits). This resulted in a total of 66,634 records for training.
  - **Validation Set**: The remaining 15% of the combined DARES data, totaling 9,391 records.

C Ablation Studies and Alternative Approaches

This appendix details the ablation studies that were conducted to validate the final system design. These experiments explored alternative approaches to key aspects of the pipeline, including preprocessing methods, Open Track data configurations, and the fundamental task formulation (classification vs. regression).

C.1 Ablation on Preprocessing: Morphological vs. Custom Pipeline

The core hypothesis of this work posits that deep morphological analysis surpasses simple surface-level normalization for Arabic readability. To test this, an alternative, custom preprocessing pipeline was implemented and evaluated. This custom method simplifies input rather than providing the linguistic enrichment of the d3tok analyzer. Its key steps include:

- **Aggressive Normalization:** Standardizes different forms of characters (e.g., Alef (أ, إ, آ) to ا, Taa Marbuta (ة) to Haa (ه)).
- **Diacritic Removal:** Strips all short vowel markings (تَشْكِيل).

This custom approach consistently yielded inferior results compared to the morphologically analyzed text. This suggests that linguistic information, such as morpheme boundaries and diacritics, preserved and added by the d3tok method, is vital for accurate text complexity assessment. Table 4 provides a direct comparison.

C.2 Ablation on Task Formulation: Classification vs. Regression

The fundamental framing of the readability assessment task was also explored. The problem can be approached as either a multi-class classification problem (predicting one of 19 discrete levels) or a regression problem (predicting a continuous score). An ablation study was conducted to evaluate the efficacy of a classification approach. As shown in Table 5, several pre-trained models were fine-tuned for sequence classification on the sentence-level task, but the regression approach ultimately yielded superior performance for this shared task.

Original Sentence	Custom Preprocessing (Tested)	d3tok Analysis (Used)
أَلَيْسَتْ هَذِهِ الْعَاطِفَةُ؟	اليست هذه العاطفه؟	أَ+ لَيْسَتْ هَذِهِ ال+ عَاطِفَةُ ؟
حَوَّلَ السَّائِقُ وَجْهَةَ فَرَسَيْهِ.	حول السائق وجهه فرسيه.	حَوَّلَ ال + سَائِقُ + وَجْهَةَ NOAN

Table 4: Comparison of the d3tok analysis (used in the final system) and the custom normalization pipeline (tested in an ablation study).

Track	Model Used	Dev (QWK)	Test (QWK)
Strict Sentence	CAMeL-Lab/readability-arabertv02-word-CE	73.31	78.20
	aubmindlab/bert-base-arabertv02	81.0	82.60
	CAMeL-Lab/readability-arabertv2-d3tok-reg	74.95	69.7
	CAMeL-Lab/bert-base-arabic-camelbert-mix-sentiment	81.60	82.70
Constrained Sentence	aubmindlab/bert-base-arabertv02	78.50	79.60
	CAMeL-Lab/readability-arabertv02-word-CE	69.0	72.20
Open Sentence	CAMeL-Lab/readability-arabertv02-word-CE	78.0	79.60

Table 5: Ablation study results for the classification approach on the sentence-level task, using the custom, non-morphological preprocessing pipeline.

D Ablation on Open Track Data Configuration

For the Open track, multiple experiments were conducted to determine the optimal mix of BAREC and DARES data, alongside ideal hyperparameters. The results, summarized in Table 6, indicate that the best configuration for the sentence-level task differed from that for the document-level task.

- **Best Sentence Performance (Exp 2):** The highest sentence-level QWK (83.9) was achieved with a lower learning rate (2e-5) over 18 epochs, using a simple concatenation of BAREC and DARES datasets for training and validation.

- **Best Document Performance (Exp 5):** The highest document-level QWK (79.2) was achieved with a higher learning rate (5e-5) and a more careful data splitting strategy. The validation set was explicitly augmented with a stratified 15% sample of the DARES data to better reflect the training distribution. This highlights the document task's sensitivity to validation set composition for robust model selection.

E Detailed Best Performing Models

This appendix presents a comprehensive breakdown of the final configurations used to achieve the best reported results. Table 7 details the specific models, training hyperparameters, data distributions, and corresponding QWK scores on both development and test sets for each track and task.

F Morphological Analysis Errors

This appendix illustrates a key challenge encountered during data augmentation: morphological analysis errors. Table 8 provides concrete examples where the CAMeL Tools d3tok analyzer failed to parse a word, inserting a NOAN (No Analysis) token. This issue arises with words not in the analyzer's vocabulary, such as uncommon proper nouns, or with words having valid but infrequent morphological forms. This introduced noise that could degrade model reliability, especially for the document-level task where a single mis-analyzed word can impact the score of an entire sentence.

ID	Training Parameters	Data Distribution	Sent. QWK	Doc. QWK
Exp 1	LR=5e-5, Epochs=6	**Train:** BAREC (54.8k) + DARES train (9.7k). **Total: 64.5k**. **Dev:** BAREC (7.3k) + DARES dev (1.4k). **Total: 8.7k**.	83.5	73.8
Exp 2	LR=2e-5, Epochs=18	Same as Exp 1.	**83.9**	76.1
Exp 3	LR=3e-5, Epochs=18	**Train:** All combined: BAREC + SAMER + DARES. **Total: 107.6k**. **Dev:** BAREC (7.3k).	83.6	74.6
Exp 4	LR=3e-5, Epochs=10	Same as Exp 1.	83.6	78.6
Exp 5	LR=5e-5, Epochs=20	**Train:** BAREC (54.8k) + 85% of merged DARES (11.8k). **Total: 66.6k**. **Dev:** BAREC (7.3k) + 15% of merged DARES (2.1k). **Total: 9.4k**.	83.0	**79.2**

Table 6: Summary of ablation experiments for the Open Track, detailing hyperparameters, data configurations, and final test set performance for both sentence and document tasks. LR refers to Learning Rate.

Track	Task	Model Used	Training Parameters	Data Distribution	Dev QWK	Test QWK
Strict	Sentences	aubmindlab/bert-base-arabertv2	Epochs=20 Learning Rate=3e-5 Warmup Ratio=0.1 Weight Decay=0.01	54845 training 7310 validation	82.30	84.20
	Document	aubmindlab/bert-base-arabertv2	Epochs=20 Learning Rate=3e-5 Warmup Ratio=0.1 Weight Decay=0.01	54845 training 7310 validation	62.37	79.90
Constrained	Sentences	CAMEL-Lab/readability-arabertv2-d3tok-reg	Epochs=8 Learning Rate=3e-5 Warmup Ratio=0.1 Weight Decay=0.01 Logging Steps=100	97874 training 7310 validation	81.0	82.9
	Document	CAMEL-Lab/readability-arabertv2-d3tok-reg	Epochs=15 Learning Rate=3e-5 Warmup Ratio=0.1 Weight Decay=0.01	97874 training 7310 validation	64.30	75.5
Open	Sentences	CAMEL-Lab/readability-arabertv2-d3tok-reg	Epochs=18 Learning Rate=2e-5 Warmup Ratio=0.1 Weight Decay=0.01	64548 training 8690 validation	82.70	83.90
	Document	CAMEL-Lab/readability-arabertv2-d3tok-reg	Epochs=20 Learning Rate=5e-5 Warmup Ratio=0.1 Weight Decay=0.01	66634 training 9391 validation	60.48	79.20

Table 7: Full details of the best performing models across all tracks and tasks.

ID	Original Sentence	D3tok Analysis
SAMER_13172	«كيف تتصرفين يا شوكار؟!»	« كيف تتصرفين يا NOAN
SAMER_15232	حول السائق وجهة فرسيه	حول ال + سائق + وجهة NOAN

Table 8: Examples of preprocessing artifacts from the SAMER corpus, where the CAMeL Tools analyzer failed to produce a morphological analysis, inserting a NOAN token instead. Failures occurred on an uncommon proper noun (شوكار), and a morphologically complex noun (فرسيه).

!MSA at BAREC Shared Task 2025: Ensembling Arabic Transformers for Readability Assessment

Mohamed Basem, Mohamed Younes, Seif Ahmed, Abdelrahman Moustafa
Faculty of Computer Science, MSA University, Egypt
{mohamed.basem1, mohamed.tarek61, seifeldein.ahmed, abdelrahman.moustafa5}
@msa.edu.eg

Abstract

We present !MSA's winning system for the BAREC 2025 Shared Task on fine-grained Arabic readability assessment, achieving first place in six of six tracks. Our approach is a confidence-weighted ensemble of four complementary transformer models (AraBERTv2, AraELECTRA, MARBERT, and CAMeLBERT) each fine-tuned with distinct loss functions to capture diverse readability signals. To tackle severe class imbalance and data scarcity, we applied weighted training, advanced preprocessing, SAMER corpus relabeling with our strongest model, and synthetic data generation via Gemini 2.5 Flash, adding 10k rare-level samples. A targeted post-processing step corrected the prediction distribution skew, delivering a 6.3% Quadratic Weighted Kappa (QWK) gain. Our system reached 87.5% QWK at the sentence level and 87.4% at the document level, demonstrating the power of model and loss diversity, confidence-informed fusion, and intelligent augmentation for robust Arabic readability prediction. [1]

1 Introduction

The BAREC 2025 Shared Task presents a formidable challenge for Arabic readability assessment. It spans six tracks (sentence and document-level across strict, constrained, and open conditions) with a fine-grained 1-19 readability scale. Predicting exact labels across such a wide range significantly increases difficulty, as even small deviations can dramatically impact metrics like Quadratic Weighted Kappa. The challenge is further compounded by severe label imbalance, where certain readability levels occur far more frequently than others, biasing models toward majority classes and making rare-level prediction unreliable. In strict and constrained tracks, limited training data amplified these issues, and in constrained settings, incorporating external datasets like SAMER (Alhafni et al., 2024; Al Khalil et al., 2020) proved non-trivial due to mismatched label distributions. Furthermore, simple scaling approaches often resulted in misalignment and minimal performance gains.

To address these challenges, we developed an ensemble framework that combines architectural and training diversity. We fine-tuned four transformer models (AraBERTv2 (Antoun et al., 2020), AraELECTRA (Antoun et al., 2021), MARBERT (Abdul-Mageed et al., 2021), and CAMeLBERT (Inoue et al., 2021)). Each model was trained with a distinct loss function (classification, regression, or ordinal). This design captures complementary signals, with outputs merged via confidence-weighted ensembling that favors more certain predictions. To mitigate data scarcity in the open tracks, we used prompt-engineered paraphrasing with the Gemini API to generate synthetic examples, and for SAMER, we relabeled instances with our best BAREC-trained model instead of relying on naive scaling.

Our approach demonstrates robustness across all track configurations of BAREC 2025, with first-place rankings in all six tracks. This success underlines the advantages of model and loss function diversity, confidence-informed fusion, and intelligent data augmentation for Arabic readability prediction, setting a strong precedent for future research in fine-grained, limited-data NLP tasks.

2 Background

2.1 Task Details

The BAREC 2025 Shared Task (Elmadani et al., 2025a) focuses on fine-grained Arabic readability assessment using the Balanced Arabic Readability Evaluation Corpus (BAREC) (Elmadani et al., 2025b). BAREC is a large-scale dataset containing over 1 million words across 68,000+ sentences and 1,900+ documents, each annotated into

[1] https://github.com/Mohamedbasem1/BAREC-2025

19 readability levels, where higher numbers indicate greater difficulty. The annotation process followed the official BAREC Annotation Guidelines (Habash et al., 2025), which define linguistic and pedagogical principles to ensure consistency and reliability in labeling.

The shared task defines two tasks: Sentence-Level & Document-Level Readability Assessment. Each one has three tracks based on permissible resources:

- **Strict Track:** Use only BAREC Corpus.
- **Constrained Track:** Use BAREC along with the SAMER Corpus and SAMER Lexicon (Alhafni et al., 2024; Al Khalil et al., 2020).
- **Open Track:** Use any additional resources or augmentation methods.

We participated in **all six tracks** across both subtasks, exploring resource-limited, resource-augmented, and fully open settings.

2.2 Related Work

Arabic readability assessment has been studied from multiple perspectives. El-Haj et al. (2024) introduced the DARES dataset for evaluating the readability of Arabic educational content, demonstrating the importance of domain-specific corpora for improving prediction accuracy. Liberato et al. (2024) proposed a hybrid approach combining handcrafted linguistic features with transformer-based models, yielding improved robustness on small or noisy datasets.

Elmadani et al. (2025b) presented BAREC, the largest balanced corpus for fine-grained Arabic readability assessment, alongside baseline systems for sentence and document-level prediction. Habash et al. (2025) detailed the annotation guidelines and methodology for BAREC, ensuring consistent application of the 19 readability levels. Alhafni et al. (2024) & Al Khalil et al. (2020) introduced the SAMER Corpus and Lexicon, designed for Arabic text simplification and multi-level difficulty annotation, which are leveraged in the Constrained Track.

Additional advances in Arabic NLP include ARBERT and MARBERT (Abdul-Mageed et al., 2021), large-scale pre-trained models that achieve state-of-the-art performance across a variety of Arabic language understanding tasks, and ensemble-based modeling for Arabic dialect identification (Khered et al., 2022), which inspired aspects of our system design.

3 System Overview

3.1 Addressing Data Imbalance

The BAREC dataset exhibits a highly imbalanced distribution across the 19 readability levels, with certain levels (e.g., 12 and 14) being far more frequent than rare levels such as 1, 18, and 19 (see Figure B.2 in Appendix). This imbalance biases models toward predicting frequent levels, which is particularly detrimental when the target metric is *Quadratic Weighted Kappa* (QWK), as misclassifying rare levels incurs a high penalty.

To mitigate this, we computed **class weights** to encourage the model to pay more attention to rare classes. The weight for each class j is calculated as:

$$w_j = \frac{n_{\text{samples}}}{n_{\text{classes}} \times n_{\text{samples in class } j}} \tag{1}$$

This formulation assigns higher weights to rarer classes and lower weights to frequent ones, reducing prediction bias and improving fairness across levels.

3.2 Model Architectures and Loss Functions

Our system builds on a diverse set of Arabic transformer models: AraBERTv2 (Antoun et al., 2020), AraELECTRA (Antoun et al., 2021), MARBERT (Abdul-Mageed et al., 2021), and CAMeLBERT (Inoue et al., 2021). These models were chosen for their strong track record in Arabic NLP benchmarks, their coverage of both Modern Standard Arabic and dialectal varieties, and their complementary pretraining objectives.

We trained multiple variants of each model using different loss formulations to capture complementary perspectives on the readability prediction problem:

- **Cross-Entropy Loss (CE)** for standard multi-class classification.
- **Mean Squared Error (MSE)** for regression over the continuous readability scale.
- **Conditional Ordinal Regression (COR)** for modeling the conditional probabilities of surpassing each readability threshold. It was implemented via the CORAL framework (Cao et al., 2019).

This diversity allowed the ensemble to leverage both discrete and continuous interpretations of the readability scale while incorporating ordinal constraints.

3.3 Constrained Track: SAMER Label Transformation

For the Constrained Track, we incorporated the SAMER Corpus (Alhafni et al., 2024), originally annotated on a 3-6 scale, into our training data. To align it with BAREC's 1-19 scale, we applied a min-max scaling transformation:

$$\text{Scaled\_Label} = \frac{\text{Label} - 3}{6 - 3} \times (19 - 1) + 1 \quad (2)$$

This transformation preserves the relative difficulty ordering while ensuring compatibility with BAREC's fine-grained labeling.

Initially, we trained our model using the scaled SAMER data and evaluated it on our BAREC test set, achieving a QWK of 50%. We then tried an alternative approach: using our best-performing BAREC-trained model directly to predict labels for the SAMER dataset on the 1-19 scale. Finally, we scaled the predictions back down to the original 3-6 SAMER range, verifying that the reverse transformation maintained accuracy within a margin of ± 0.5. This approach significantly improved results.

3.4 Open Track: Data Augmentation with Gemini 2.5 Flash

In the Open Track, we expanded our training corpus using `Gemini 2.5 Flash`. As seen in Figure C.1, few-shot prompting with high-quality examples from BAREC was utilized to generate rephrasings and additional readability-graded sentences, resulting in approximately 10k new samples. This augmentation improved coverage for rare and boundary-level readability cases.

3.5 Ensembling Strategy

Model predictions were combined using a **confidence-weighted averaging scheme**:

$$W = \frac{\sum_{i=1}^{n} p_i \, c_i}{\sum_{i=1}^{n} c_i} \quad (3)$$

where p_i is the predicted readability score from model i, c_i is the model confidence (derived from softmax probabilities for classification and inverse variance for regression), and n is the number of models. This approach prioritized more certain predictions, improving robustness across evaluation tracks.

For specific cases, a secondary method combined two predictions as:

$$E = \begin{cases} \max(p_1, p_2), & \text{if } |p_1 - p_2| = 1 \\ \frac{p_1 + p_2}{2}, & \text{otherwise} \end{cases} \quad (4)$$

This rule-based adjustment handled borderline cases where one-point differences significantly impact evaluation metrics.

3.6 Document-Level Prediction Aggregation

While our models initially produce sentence-level predictions, the document-level track requires aggregating these predictions to the document level. Following guidance from the task organizers, we extract document IDs using the first 7 characters of each sentence ID and apply a **maximum aggregation rule**:

$$R_{\text{doc}} = \max_{s \in S_{\text{doc}}} R_s \quad (5)$$

where R_{doc} is the final document readability prediction, S_{doc} represents all sentences in a document, and R_s is the sentence-level prediction.

This approach, recommended by the organizers, assumes a document's readability is constrained by its most challenging sentences.

4 Experimental Setup

4.1 Data and Splits

We use the BAREC dataset with Arabic texts labeled on a 19-level readability scale, following the official train, dev, and test splits. In the Strict Track, only BAREC training data was used. In the Constrained Track, we added the SAMER dataset relabeled to 19 levels using our best BAREC model. In the Open Track, we further augmented training with synthetic samples from Gemini 2.5 Flash.

4.2 Preprocessing Pipeline

Our pipeline (Figure B.1) includes:

1. **Data cleaning:** Removing redundant punctuation, normalizing special characters, and trimming extra spaces via regular expressions.

2. **Morphological tokenization:** Using `D3TOK` from CAMeL Tools (Obeid et al., 2020) to preserve morphological segments.

3. **Class imbalance handling:** Applying inverse-frequency class weights to improve predictions for rare levels.

4.3 Model Training Configuration

We fine-tuned four pretrained transformer-based language models with different loss functions. Training was conducted using the Hugging Face `Transformers` library with the hyper-parameters from Table B.3. All experiments ran on L40s GPUs with mixed-precision acceleration (`torch.cuda.amp`).

4.4 Evaluation Metrics

We evaluate on both development and official test sets using :

- **Quadratic Weighted Kappa (QWK)** - primary metric, penalizing distant misclassifications more heavily.

- **Accuracy (Acc)** - reported for 19, 7, 5, and 3 predicted label levels.

- **Adjacent Accuracy (±1 Acc19)** - off-by-one tolerance.

- **Average Distance (Dist)** - measures the average absolute distance between predicted and true labels.

5 Result

Table A.1 compares the QWK performance of individual model variants against their ensembles. Singular models achieved QWK scores ranging from 81.0% to 84.8%, with MARBERT+COR achieving the highest among single models. When combined into ensembles, performance consistently improved, with our best ensemble achieving 87.5% QWK, representing a notable gain over the best single model.

An important insight came from analyzing prediction distributions in the document-level tracks. Figure A.2 shows the label frequency distributions before (left) and after (right) a post-processing adjustment. Initially, there were no predictions for label 10, and the distribution was skewed due to our document-level aggregation method, which involved taking the average readability score among document and applying a ceiling function to round decimals up. This approach, when document-level predictions were close in value, sometimes produced unrealistic final document scores.

Upon realizing this issue, we experimented with replacing the ceiling operation with a flooring operation in such borderline cases. In parallel, we also addressed another skew in the distribution, the appearance of label 15 with disproportionately high frequency. To mitigate this, we introduced a heuristic in the ensemble post-processing: if any of the models predicted labels 16 or 17 for a document, we overrode the averaged ensemble prediction with that higher label.

Both of these adjustments contributed to a substantial performance boost, increasing QWK result by 6.3%. The changes not only improved label coverage (including the introduction of label 10 predictions) but also redistributed predictions more evenly across higher readability levels.

Table 1 reports our performance across six tracks in the Sentence-Level and Document-Level tasks, under Strict, Constrained, and Open settings.

At the Sentence Level, our best QWK scores reached 87.5% (Strict, Run 1), 86.6% (Constrained, Run 1), and 86.4% (Open, Run 1), securing 1st place in all three tracks. These results show consistent top performance across multiple runs, with very close QWK values among them, indicating stability.

At the Document Level, our highest QWK scores were 87.4% (Strict, Run 1), 84.3% (Constrained, Run 1), and 82.2% (Open, Run 1). Again, we ranked 1st place in both Strict, Constrained and Open settings. The results also show that the Strict track generally yielded higher QWK and accuracy scores than Constrained and Open.

6 Conclusion

This work presented an ensemble-based system for Arabic readability assessment in the BAREC 2025 Shared Task. By combining four transformer models (AraBERTv2, AraELECTRA, MARBERT, CAMeLBERT) with diverse loss functions, confidence-weighted ensembling, and data augmentation via Gemini 2.5 Flash.

Our system secured **first place in five of six tracks**, achieving QWK scores of **87.5%** (sentence-level) and **87.4%** (document-level). Post-processing adjustments to correct distribution skew further boosted performance by **6.3%**,

Task	Track	Run	QWK	Acc19	Acc7	Acc5	Acc3	±1 Acc19	Rank
Sentence Level	Strict	Run 1	**87.5**	43.5%	64.1%	69.6%	76.2%	76.7%	1st / 39
		Run 2	87.4	42.5%	63.5%	69.2%	76.1%	76.5%	
		Run 3	87.2	40.9%	63.4%	69.1%	76.2%	76.1%	
	Constrained	Run 1	**86.6**	44.9%	63.0%	68.7%	75.6%	75.4%	1st / 20
		Run 2	86.5	42.6%	61.5%	67.3%	74.5%	75.6%	
		Run 3	86.2	39.2%	60.9%	67.4%	74.7%	74.5%	
	Open	Run 1	**86.4**	41.3%	61.7%	67.3%	74.5%	75.1%	1st / 22
		Run 2	86.3	41.5%	60.9%	66.8%	75.0%	73.8%	
		Run 3	86.1	40.0%	61.4%	67.4%	74.6%	74.8%	
Document Level	Strict	Run 1	**87.4**	52.0%	81.0%	81.0%	93.0%	94.0%	1st / 27
		Run 2	80.2	42.0%	68.0%	68.0%	86.0%	89.0%	
		Run 3	79.3	41.0%	67.0%	67.0%	86.0%	88.0%	
	Constrained	Run 1	**84.3**	48.0%	77.0%	77.0%	94.0%	91.0%	1st / 22
		Run 2	82.3	47.0%	72.0%	72.0%	89.0%	86.0%	
		Run 3	78.9	41.0%	67.0%	68.0%	88.0%	86.0%	
	Open	Run 1	**82.2**	50.0%	70.0%	70.0%	89.0%	86.0%	1st / 19
		Run 2	78.6	42.0%	67.0%	67.0%	86.0%	86.0%	
		Run 3	76.2	39.0%	63.0%	63.0%	83.0%	84.0%	

Table 1: Top 3 performances across each tracks using Quadratic Weighted Kappa (QWK), Accuracy at multiple levels (Acc19/7/5/3), Off-by-1 Accuracy (±1 Acc19), and Average Distance (Dist). Along with the rank achieved in each track / Number of participants.

underscoring the value of model diversity and confidence-guided ensembling for fine-grained Arabic readability prediction.

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.

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2025. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. 2019. CORAL: Rank-consistent ordinal regression for neural networks. *arXiv preprint arXiv:1901.07884*.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Mo El-Haj, Sultan Almujaiwel, Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2024. DARES: Dataset for Arabic readability estimation of school materials. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 103–113, Torino, Italia. ELRA and ICCL.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. 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.

Abdullah Khered, Ingy Abdelhalim Abdelhalim, and Riza Batista-Navarro. 2022. Building an ensemble of transformer models for Arabic dialect classification and sentiment analysis. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 479–484, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. 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.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733, version 2.

A Model Performance Analysis

A.1 Ensemble Results Comparison

AraBERT			AraELECTRA			CamelBERT			MarBERT			Metrics
CE	REG	COR	CE	REG	COR	CE	REG	COR	CE	REG	COR	QWK
Singular Models												
									✓			81.0%
										✓		83.0%
								✓				83.1%
	✓											84.1%
							✓					84.5%
				✓								84.8%
Ensembles												
				✓				✓				85.3%
		✓		✓				✓				86.2%
✓		✓		✓		✓	✓					86.9%
✓		✓		✓		✓	✓				✓	**87.5%**

Table 2: Ensemble model Quadratic Weighted Kappa results comparison split into Singular Models and Ensembles.

A.2 Prediction Distribution Results

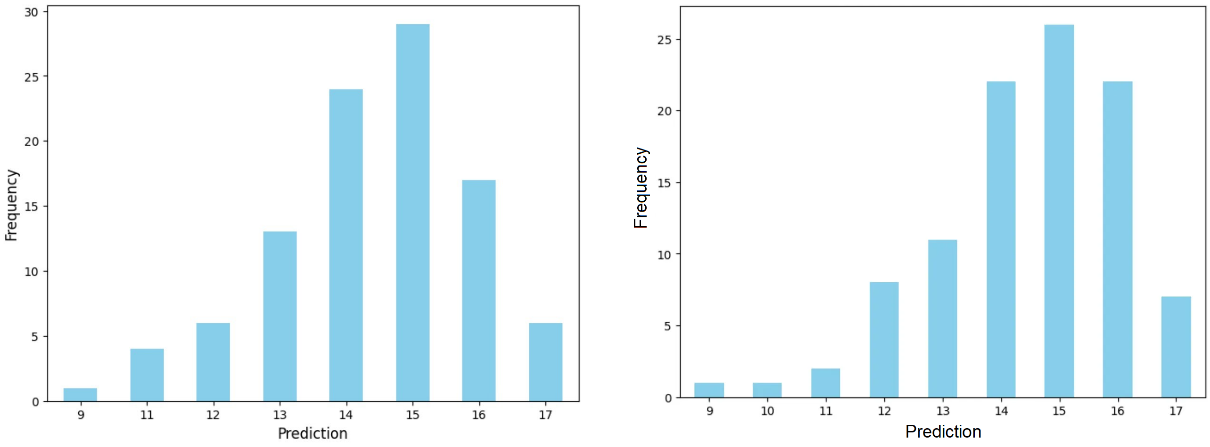


Figure 1: Graphs of Distribution of Predictions before (left) and after (right) adjusting skewness.

B System Architecture and Configuration

B.1 Architecture Overview

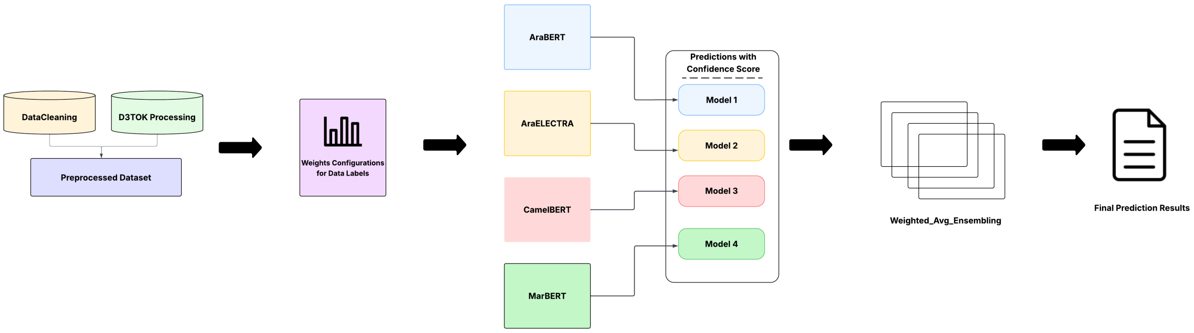


Figure 2: System Architecture Diagram

B.2 Dataset Distribution

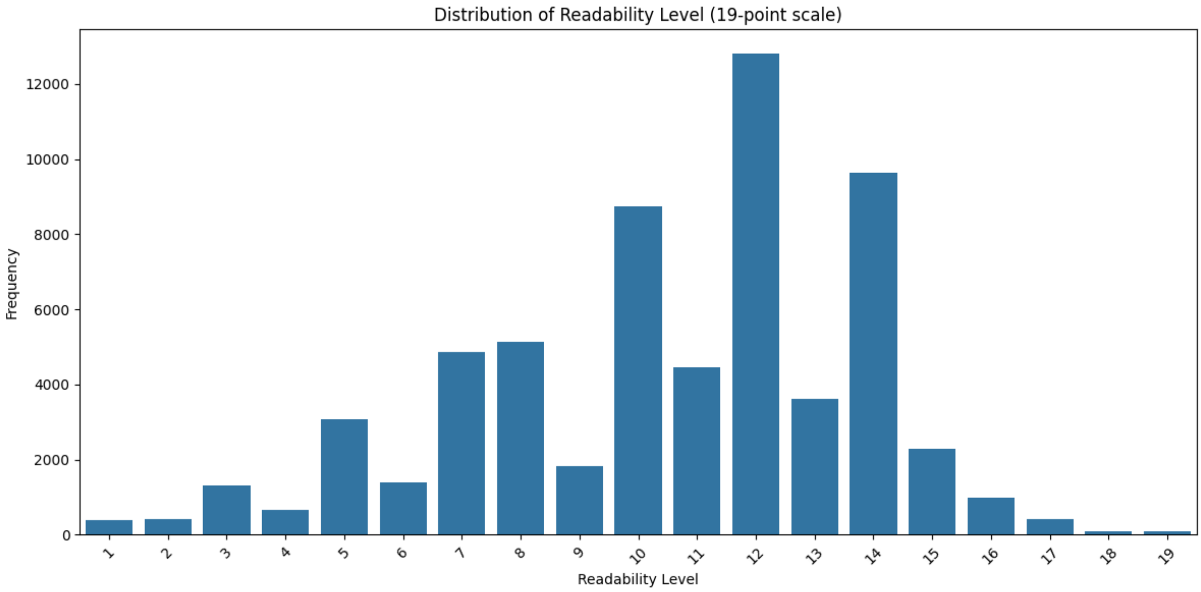


Figure 3: Distribution of Readability Levels across the dataset.

B.3 Training Configuration

Hyper-parameters	Values
Batch Size	16
Learning Rate	$2X10^-5$
Epochs	5
Optimizer	AdamW
Callbacks	EarlyStopping

Table 3: Hyper-parameters used in model training.

C Data Augmentation Details

C.1 LLM Prompt Template

Few-Shot LLM Prompt

المهمة: أعد صياغة الجملة العربية مع الحفاظ على:
- نفس عدد الكلمات
- نفس علامة الترقيم
- نفس مستوى القراءة: {readability_desc}
- نفس المعنى العام
- استخدام مفردات وتراكيب بنفس مستوى الصعوبة

مثال 1:
الجملة الأصلية: "ماجد"
عدد الكلمات: 1
مستوى القراءة: مستوى أساسي جداً - كلمات بسيطة ومألوفة
علامة الترقيم: ""

الجملة المعاد صياغتها: "فهد"

مثال 2:
الجملة الأصلية: "السنة الثامنة"
عدد الكلمات: 2
مستوى القراءة: مستوى متوسط مبكر - كلمات متنوعة وتراكيب متوسطة
علامة الترقيم: ""

الجملة المعاد صياغتها: "العام الثامن"

مثال 3:
الجملة الأصلية: "الأربعاء 21 يناير 1987"
عدد الكلمات: 4
مستوى القراءة: مستوى فوق المتوسط - مفردات متقدمة وتراكيب معقدة
علامة الترقيم: ""

الجملة المعاد صياغتها: "يوم الأربعاء 21 كانون الثاني 1987"

======================
الجملة المطلوب إعادة صياغتها:

الجملة الأصلية: "{sentence}"
عدد الكلمات: {word_count}
مستوى القراءة: {readability_desc}
علامة الترقيم: "{punctuation}"

المطلوب:
أعد صياغة الجملة مع الحفاظ على نفس عدد الكلمات ({word_count})، ونفس مستوى القراءة، ونفس المعنى، ونفس علامة الترقيم "{punctuation}" في النهاية.

قدم فقط الجملة المعاد صياغتها بدون أي شرح أو تعليق إضافي.

Qais at BAREC Shared Task 2025: A Fine-Grained Approach for Arabic Readability Classification using a Pre-trained Model.

Samar Ahmed[1]
samar.sass6@gmail.com

[1]NAMAA, Riyadh, Saudi Arabia

Abstract

In this paper, the results are presented within the context of the BAREC 2025 Shared Task (Elmadani et al., 2025a; Habash et al., 2025; Elmadani et al., 2025b) for Arabic text readability prediction. Participation in both the strict and open tracks achieved QWK scores of 82.5% and 83%, respectively. The proposed approach employs a 19-level fine-grained classification framework at the sentence level, leveraging the BAREC dataset (Elmadani et al., 2025a; Habash et al., 2025; Elmadani et al., 2025b) and transformer based $AraBERT$ models. To address class imbalance, underrepresented levels were augmented with additional samples. By incorporating rich linguistic and structural features, including morphology, syntax, and vocabulary, the system surpasses less fine-grained methods in precision and reliability.

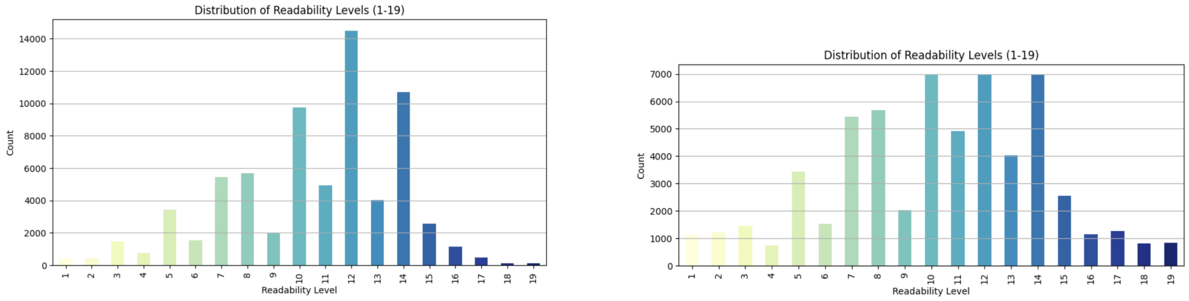
1 Introduction

Readability indicates to the ease with which someone can understand a particular text or sentence (Nassiri et al., 2022). Although the majority of early research in this area focused on English due to the abundance of rich and extensive datasets, readability evaluation for Arabic and other languages has gained attention in recent years. However, the Arabic language presents unique challenges, such as the lack of annotated datasets and the complexities of its syntactic and morphological structure. Readability is therefore a critical aspect of NLP, with practical implications across domains like education, public communication, and digital platforms, where improving text clarity enhances understanding for both native speakers and language learners. A number of studies have attempted to assess Arabic readability using various metrics and linguistic levels. For example, (El-Haj and Rayson, 2016) counts stressed, long, and short syllables to measure readability. This technique is a good starting point, but it falls short of effectively expressing the complex details of Arabic syntax and morphology. The BAREC dataset (Elmadani et al., 2025a; Habash et al., 2025) addresses some of these limitations by incorporating a wider range of linguistic and structural features such as spelling, word count, morphology, syntax, vocabulary, and content to improve classification accuracy. Previous studies have examined readability at various linguistic levels, including the sentence, document, word, and token levels, and have employed different scales, ranging from 3 to 7 levels, such as (Al Khalil et al., 2020) and (Hazim et al., 2022). However, no prior work has systematically investigated the potential of fine-grained readability levels for Arabic, particularly when combined with advanced transformer based language models at the sentence level. In this study, the limitation of broad-scale readability measures is addressed by employing a fine-grained 19-level classification system derived from the BAREC dataset. This framework is applied at the sentence level using large-scale Arabic language models based on $AraBERT$. By combining a fine-grained readability scale with advanced transformer based language models, the proposed approach aims to produce more accurate and reliable readability estimates for Arabic texts. This contribution not only expands the methodological landscape of Arabic readability assessment but also provides a scalable foundation for educational, institutional, and technological applications requiring precise control over text complexity. The rest of this paper is organized as follows: Section 2 reviews related work on Arabic readability, Section 3 describes the methodology, Section 4 presents the model results, Section 5 offers a discussion, and Section 6 provides an error analysis.

2 Related work

Focusing on Arabic readability research, researchers have made significant efforts to address

Proceedings of The Third Arabic Natural Language Processing Conference, pages 306–311
November 8-9, 2025 ©2025 Association for Computational Linguistics



(a) Distribution before balancing

(b) Distribution after balancing

Figure 1: Distribution of dataset before and after balancing

the scarcity of data by building datasets to measure text difficulty, supporting the Arabic NLP community. One notable contribution is Al Khalil et al. (2020) and Alhafni et al. (2025), the Simplification of Arabic Masterpieces for Extensive Reading (SAMER) project, which presents a five-level readability lexicon for Modern Standard Arabic, manually annotated by language professionals from three Arab regions. The lexicon was built from news articles and literary texts. Following this, Al-Twairesh et al. (2016) introduced MADAD, a tool based on collecting readability annotations on Arabic texts at the sentence and paragraph levels using pairwise and direct rating methods, helping to fill the gap in Arabic readability data. Subsequent research, such as Elmadani et al. (2025a); Habash et al. (2025), developed a large and reliable dataset for assessing Arabic text readability at multiple granularities, fine-tuning $AraBERT$ to establish a baseline for sentence-level classification. Studies leveraging the SAMER dataset have used varied approaches:Liberato et al. (2024) assessed readability with methods ranging from rule-based to pretrained language models, and Hazim et al. (2022) presented a Google Docs add-on for automatic Arabic word-level readability visualization, providing difficulty assessment, substitution suggestions, and foundational resources such as a graded readability lexicon and a parallel corpus.

3 Methodology

3.1 BAREC Dataset

The BAREC dataset (Elmadani et al., 2025a; Habash et al., 2025; Elmadani et al., 2025b) contains 69,441 Arabic sentences (more than 1 million words) from various genres and audiences, annotated across 19 readability levels from kindergarten to postgraduate, following the Inspired by the Taha/Arabi21 (Taha-Thomure, 2017). Annotations are performed manually, with high agreement between annotators (Quadratic Weighted Kappa = 81.8%), ensuring data quality. It is openly available and benchmarked using multiple readability assessment methods, supporting research and educational applications in Arabic readability.

3.1.1 Dataset for Strict track

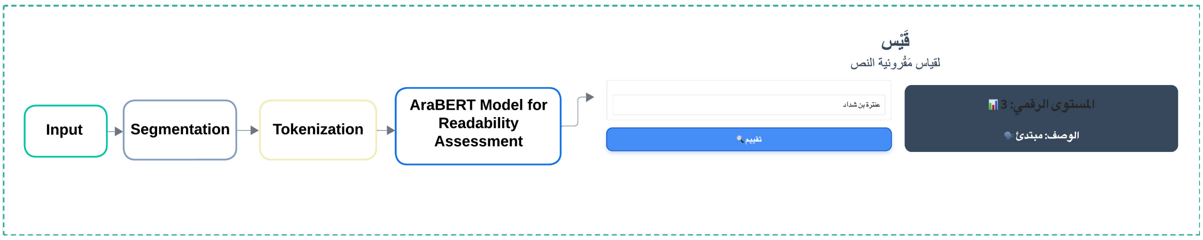
For this track, the original BAREC dataset was used without any modifications, and no data augmentation was applied. The dataset consisted of 69,441 rows, with 80% allocated for training, 10% for development, and 10% for testing. Furthermore, BAREC provided a blind test set of 3,420 cases.

3.1.2 Dataset for Open track

In the Open track, we extended the training data by generating synthetic sentence-level examples using ChatGPT-based augmentation to improve model generalization across underrepresented readability levels. The original dataset was highly imbalanced, with some classes significantly overrepresented while others had very few samples. To address this, both up-sampling and down-sampling techniques were applied. Specifically, levels 1, 2, 17, 18, and 19 were up-sampled using GPT-generated data, whereas levels 10, 12, and 14 were down-sampled to 7,000 instances. This number was selected because it closely matches the size of the dataset's largest class after removing the three over-represented categories, thereby helping to balance the data distribution, as illustrated in Figure1a and Figure 1b. The final dataset consisted of 59,236 rows.

3.2 Model

$AraBERT$, a BERT-based pre-trained language model developed specifically for Arabic (Antoun et al., 2020), was introduced to address



قَيْس

لقياس مَقْروئية النص

عنترة بن شداد

المستوى الرقمي: 3 📊

الوصف: مبتدئ 🗣

Figure 2: Flow of Qais model

the limitations of multilingual models by providing an architecture optimized for Arabic NLP tasks. It has achieved notable improvements in tasks such as Sentiment Analysis, Named Entity Recognition, and Question Answering. In the present work, two variants were utilized: aubmindlab/bert-arabertv2 and aubmindlab/bert-large-arabertv2, with aubmindlab/bert-arabertv2 selected as the primary model due to its stronger performance. Figure 2 illustrates the flow diagram for classifying Arabic texts according to readability levels. The process starts with entering the sentence, followed by segmentation. The segmented data is then tokenized and fed into the model, which has been trained on segmented data to enhance accuracy. Finally, the model produces a readability classification for the input sentence.

3.3 Hyperparameters

As part of hyperparameter optimization, the models were trained using NVIDIA A100 and T4 GPUs in Google Colab. The learning rates were set to either 2e-5 or 5e-5, with a weight decay of 0.01 to mitigate overfitting. Batch sizes were configured to 4 or 8, depending on the model's complexity and resource requirements. Maximum number of epochs was set to 20, and the AdamW optimizer, which is used by default in $AraBERT$, was employed during training.

4 Results

This task include a readability assessment, which evaluates both tracks using multiple metrics (Elmadani et al., 2025a; Habash et al., 2025; Elmadani et al., 2025b). Quadratic weighted Kappa (QWK) measures the agreement between predictions and accurate labels, with higher penalties for larger errors. It is the primary evaluation metric. Accuracy (Acc) is the percentage of exact matches between predictions and accurate labels using a 19-level scale (Acc19). Simplified versions include Acc7, Acc5, and Acc3, where the 19 levels are grouped into 7, 5, or 3 categories. Adjacent Accuracy (±1 Acc19) counts predictions as correct if they are exactly right or within ±1 level of the actual label. The average distance (dist) or mean absolute error (MAE) measures the average absolute difference between the predicted and actual labels. In the Readability Assessment task, results are reported for two evaluation tracks: Sentence-level Strict and Sentence-level Open. In the first track, the original BAREC dataset was used without any modifications, a QWK score of 82.5%. In the second track, data augmentation techniques, including up-sampling and down-sampling, were applied to the BAREC dataset, resulting in a slightly improved QWK score of 83.0%, as shown in the Table 1. These findings underscore the significant impact of data balancing on model performance. Two variants from the $AraBERT$ series were experimented with: aubmindlab/bert-base-arabertv2 and aubmindlab/bert-large-arabertv2. Initial experiments with arabertv2-base consistently yielded high performance, with QWK scores ranging between 80 and 83 across both tracks. In contrast, improving to the larger $AraBERTv2$ model resulted in reduced accuracy, with QWK scores ranging from 70% to 78%. Different learning rates (2e-5 and 5e-5) were also investigated, with 2e-5 consistently yielding better outcomes. Coral Loss, which preserves the ordinal character of labels by penalising predictions based on their distance from the actual label, was also investigated. However, when Coral Loss was applied, accuracy decreased.

5 Discussion

In this study, it was observed that the base-arabertv2 model, when trained on the BAREC dataset in both tracks, outperformed the large-arabertv2 model. This is likely because the large-arabertv2 model requires a larger dataset and greater computational resources. In the open track, a slight improvement over the strict track was recorded, which can be attributed to the signifi-

Track	QWK	±1 Acc	Acc19	Dis	Acc 7	Acc 5	Acc 3
Sentence-level Strict	82.5	54.8	71.8	1.1	65.1	69.5	75.3
Sentence-level Open	83.0	54.2	71.8	1.1	66.0	70.0	75.8

Table 1: Performance results Readability Assessment for both tracks

cant class imbalance in the dataset. Furthermore, there was substantial variance across the 19 readability levels: the initial and final levels contained far fewer samples, While the middle levels had relatively more data. This uneven distribution hurt the model's overall performance. To address this issue, class weights were applied in the strict track to reduce the impact of the severe imbalance. In the open track, the data was manually balanced and reduced, with adjustments made to extreme classes. However, the improvement achieved was not substantial, likely because class weights had already been applied to mitigate the imbalance. Upon examining the dataset, it was also found that some rows were duplicated and contained unfamiliar words, such as كونغ فو *kwn fw* which appeared frequently. Although written in Arabic script, كونغ فو *kwn fw* is a foreign term, and its repetition could potentially hinder the model's ability to interpret and classify inputs accurately. For example, the term might be classified as a high difficulty word, while in reality, it is simply a proper noun commonly used in western contexts. It was also noted that some sentences contained non-Arabic words written in Arabic characters. Such issues may reduce the clarity of the dataset's texts and hinder the overall performance of the model. Although the last few levels (17, 18, and 19) are highly similar, this did not cause significant confusion for the model, as their difficulty is very close. Merging these levels into a single unified level might have yielded slightly better results than keeping them separate. In contrast, differences between other levels appeared more distinct and beneficial, and it is likely that levels containing more data were classified with greater accuracy.

6 Error analysis

To better understand the model's performance in the strict and open tracks, a manual analysis was conducted on more than ten randomly selected sentences with divergent readability labels. The analysis revealed that both tracks produced competitive results, with minimal differences in overall performance. However, specific error patterns were observed. When an Arabic word contained or was attached to numbers, the model occasionally generated inconsistent readability predictions. For example, in the sentence 208 ماجد *mAjd*, which represents only a name, the expected classification was level 1; the trained model, however, assigned it level 3 in both tracks. Although such cases may not significantly affect performance, numbers can sometimes alter the contextual interpretation. In the sentence 10 جمادى الأولى *jmAdý AlÂwlý* 1440 هـ *h*, the model assigned a score of 13 in both tracks, likely due to numerical elements introducing classification confusion. This misclassification is particularly problematic as the actual difficulty level of the sentence is beginner-level. Another difficulty that has been noticed is that words written in Arabic script are derived from English. These phrases frequently obtained excessively high readability scores, despite their difficulty correlating more closely with the first or second levels of difficulty. Such misclassification can result in content incompatibilities with the intended audience. At higher levels, notably above level 10, the model demonstrated improved classification accuracy, with errors becoming less common and severe. This improvement can be attributed to the use of better syntactic and lexical patterns in larger phrases, which are less likely to contain numbers or symbols that could interfere with the model's classification process.

7 Conclusion and Future Work

In this work, a fine-tuned $AraBERT$ model was presented for the BAREC shared task in the strict and open tracks, targeting Arabic sentence readability assessment. The results, while satisfactory, indicate potential for further improvement. Future work will begin with traditional machine learning approaches and progress towards deep learning methods, ultimately leveraging pre-trained models, alongside enhanced data cleaning, class balancing, and class merging. The system is envisioned to be deployed as a web-based tool for the Arabic.

Limitations

Resource constraints on Google Colab Pro limited experimentation with larger datasets and models, with restricted RAM causing occasional training crashes. To mitigate this issue, batch sizes were reduced; however, future experiments will require access to larger computing resources to fully realize the model's potential.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Nora Al-Twairesh, Abeer Al-Dayel, Hend Al-Khalifa, Maha Al-Yahya, Sinaa Alageel, Nora Abanmy, and Nouf Al-Shenaifi. 2016. Madad: a readability annotation tool for arabic text. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4093–4097.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2025. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

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.

Mahmoud El-Haj and Paul Rayson. 2016. OSMAN — a novel Arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255, Portorož, Slovenia. European Language Resources Association (ELRA).

Khalid Elmadani, Nizar Habash, and Hanada Taha. 2025a. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025b. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Naoual Nassiri, Abdelhak Lakhouaja, and Violetta Cavalli-Sforza. 2022. Arabic l2 readability assessment: Dimensionality reduction study. *Journal of King Saud University-Computer and Information Sciences*, 34(6):3789–3799.

Hanada Taha-Thomure. 2017. *Arabic Language Text Leveling*. Educational Book House.

Appendix A: Readability Dimensions Used for Sentence Generation

In my experiments, I provided GPT with the six dimensions from the BAREC readability framework (Elmadani et al., 2025a; Habash et al., 2025) and asked it to follow them when generating sentences at different readability levels. These dimensions are briefly described below:

1. **Word Count:** Measured by counting unique printed words (punctuation and diacritics ignored). This feature is constrained to a maximum of 20 words up to level 11 (Kaf).

2. **Orthography and Phonology:** Focused on word length (syllable count) and special letters such as hamzas. Final diacritics are ignored (words are read in pause form).

3. **Morphology:** Included derivation and inflection (e.g., tense, aspect, number). Simpler forms (e.g., present tense before past, singular before plural) appear at lower levels. This feature is used up to level 13 (Mim).

4. **Syntactic Structures:** Tracked sentence complexity, ranging from single words (level 1 –

Alif) to more complex structures. Applied up to level 15 (Seen).

5. **Vocabulary:** Central across all levels. Shared words across dialects and Modern Standard Arabic appear in easier levels, while technical terms are introduced in higher levels.

6. **Ideas and Content:** Evaluated required prior knowledge, symbolic decoding, and conceptual connections. Progression moves from familiar ideas to specialized knowledge, and from literal meanings to abstract concepts.

These dimensions guided the construction of sentence examples used in our readability experiments.

mucAI at BAREC Shared Task 2025: Towards Uncertainty Aware Arabic Readability Assessment

Ahmed Abdou
Independent Researcher. Munich, Germany
ahmedabdou1789@gmail.com

Abstract

We present a simple, model-agnostic post-processing technique for fine-grained Arabic readability classification in the BAREC 2025 Shared Task (19 ordinal levels). Our method applies conformal prediction to generate prediction sets with coverage guarantees, then computes weighted averages using softmax-renormalized probabilities over the conformal sets. This uncertainty-aware decoding improves Quadratic Weighted Kappa (QWK) by reducing high-penalty misclassifications to nearer levels. Our approach shows consistent QWK improvements of 1-3 points across different base models. In the strict track, our submission achieves QWK scores of 84.9%(test) and 85.7% (blind test) for sentence level, and 73.3% for document level. For Arabic educational assessment, this enables human reviewers to focus on a handful of plausible levels, combining statistical guarantees with practical usability.

1 Introduction

Automatic readability assessment estimates how difficult a text will be for a target audience, a task essential for the design and advancement of pedagogically oriented NLP applications(Collins-Thompson and Callan, 2004; Xia et al., 2016). In Arabic, this problem is particularly challenging due to morphological richness, and orthographic variation (Liberato et al., 2024; Benajiba and Rosso, 2008). Recent work has advanced Arabic readability assessment through modeling and datasets (Saddiki et al., 2018; Alhafni et al., 2024; Elmadani et al., 2025a; Habash et al., 2025). Most recently, the BAREC corpus (Elmadani et al., 2025a) which offers 19 fine-grained levels. Nevertheless, even state-of-the-art models like AraBERT-v2 (Antoun et al., 2020) remain prone to large-gap misclassifications and offer no principled means of quantifying prediction uncertainty. We address this by integrating conformal prediction (Vovk et al., 2005) to produce statistically valid prediction sets and uncertainty-guided final predictions, reducing high-penalty errors and enabling compact, interpretable outputs for human-in-the-loop educational use. On the BAREC 2025 Shared Task, our method consistently improves QWK across base models, reaching 84.9% on the test set and 85.7% on the blind test at the sentence level, and 73.3% on the blind test at the document level. Beyond leaderboard improvements, our method provides interpretable prediction sets and uncertainty estimates that enable more reliable readability assessment. Our implementation is open-sourced for reproducibility[1].

2 Background

2.1 Task and Data

The BAREC Shared Task 2025 (Elmadani et al., 2025b) targets fine-grained Arabic readability assessment across 19 ordered levels. The task builds on the BAREC corpus (Elmadani et al., 2025a), a manually annotated dataset containing over 69,000 sentences and more than one million words. The corpus provides mappings to multiple granularities (3, 5, and 7 readability levels); for detailed annotation guidelines, we refer readers to (Habash et al., 2025). We participated in both sentence-level and document-level variants of the strict track, where participants are restricted to using only the BAREC corpus for training. In the document-level task, a document's overall readability level is determined by its most difficult sentence. Given the ordinal nature of readability levels, the main evaluation metric is Quadratic Weighted Cohen's Kappa (QWK), which penalizes larger misclassifications more heavily (Cohen, 1968). This reflects the educational goal of avoiding assignments far from a student's level. We also report exact accuracy, ad-

[1] https://github.com/AhmedAbdel-Aal/mucAI-at-BAREC_2025

Proceedings of The Third Arabic Natural Language Processing Conference, pages 312–319
November 8-9, 2025 ©2025 Association for Computational Linguistics

jacent accuracy (±1 of true label), Mean Absolute Error (MAE), and coarse-grained variants Acc7, Acc5, and Acc3, which collapse the 19 levels into 7, 5, and 3 bins. The shared task provides standard splits: training (54.8k), development (7.3k), test (7.3k), and blind test (3.4k), with the first three publicly available[2].

2.2 Conformal Prediction

Conformal Prediction (CP) (Vovk et al., 2005; Papadopoulos et al., 2002) is a model-agnostic method that converts single predictions into prediction sets with statistical guarantees. Rather than predicting "this text is Level 9", CP produces "this text is likely Level 7, 8, 9, 10, or 11". Given a target miscoverage rate α, CP guarantees that the true label appears in the prediction set with probability at least $1 - \alpha$:

$$P\big(Y \in C(X)\big) \geq 1 - \alpha \tag{1}$$

where $C(X)$ is the predicted set for input X and Y is the true label. The method works by using a calibration set, data not seen during training, to learn how "unusual" different labels are for given inputs. This unusualness is captured by a nonconformity score $s(x, y)$: higher scores mean label y is less plausible for input x (more in appendix A.1.). CP then sets a threshold $\hat{\tau}$ which is chosen as the $(1 - \alpha)(n + 1)$-quantile of these scores in the calibration set, ensuring the coverage guarantee. For any new input x, the prediction set includes all labels below this threshold:

$$C(x) = \{y \in \mathcal{Y} : s(x, y) \leq \hat{\tau}\} \tag{2}$$

3 Method

We use AraBERT-v2 (Antoun et al., 2020) as the backbone, following the strongest BAREC baselines (Elmadani et al., 2025a). The original benchmark reports four preprocessing pipelines based on CAMeL tools (Obeid et al., 2020) (Word, Lex, D3Lex, D3Tok) but we could not run the CAMeL D3 analyzer in our environment. Because BAREC releases the dev/test sentences already preprocessed with these pipelines, we include them for comparison. For the blind split, however, only raw text is provided; we therefore adopt AraBERT's recommended Farasa segmentation (Abdelali et al., 2016). For training objectives, we replicate the benchmark baselines: Cross-Entropy (CE) and Earth Mover's Distance (EMD) (Hou et al., 2017), and an ordinal Regression variant. Our addition is a Focal-loss objective (Lin et al., 2017) tailored to the long-tailed 19-level label distribution; we report it alongside the baselines and simple ensembles: probability averaging, and majority voting.

Our post-processing approach combines conformal prediction with expected value decoding. We first generate prediction sets with coverage guarantees, then produce final predictions by averaging within these sets. We apply CP only to the probabilistic classifiers (CE/EMD/Focal); the Regression head is reported as point predictions only.

Prerequisites and Notation. Let $\mathcal{Y} = \{1, ..., 19\}$ denote the ordered labels. A trained classifier produces posterior probabilities $p(y \mid x)$ for input x. For any x, we build form a conformal prediction set $C(x) \subseteq \mathcal{Y}$ and then decode to a single label.

Calibration and Tuning Protocol. We split the official development set into two stratified halves: a calibration split (*dev-cal*) for learning conformal thresholds, and a tuning split (*dev-tune*) for hyperparameter selection and evaluation. See the split details in Table 5 in Appendix A.5.

Set Construction. We evaluate three standard nonconformity score functions for multiclass conformal prediction: naïve (inverse-probability), APS (Adaptive Prediction Sets) (Romano et al., 2020), and RAPS (Regularized APS) (Angelopoulos et al., 2020).

Renormalization within the set. We first renormalize probabilities within the conformal set

$$p_C(y \mid x) \;=\; \frac{p(y \mid x)}{\sum_{j \in C(x)} p(j \mid x)} \quad \text{for } y \in C(x).$$

We then predict the rounded posterior mean

$$\hat{y}(x) \;=\; \text{round}\left(\sum_{y \in C(x)} y \, p_C(y \mid x) \right).$$

The choice of weighted mean is motivated by its role as the Bayes-optimal point estimator under quadratic loss. While this is not strictly optimal for our discrete classification setting, we employ it as a computationally simple heuristic that aligns with the quadratic penalty structure of the primary evaluation metric (QWK). For the document-level

[2] BAREC corpus shared-task-2025

track, we applied our best-performing sentence-level model to all sentences in a document and assigned the document's readability as the maximum predicted level across its sentences, following the shared task definition. We report the full experimental setup in appendix A.2.

4 Results

Dev/test results demonstrate that clitic-aware preprocessing substantially improves performance: Farasa and D3Tok consistently outperform word-level and lexical baselines, with Farasa achieving the best QWK scores under CE, EMD, and regression losses, and on par under Focal loss. Given Farasa's consistent performance across dev/test splits and its availability as the only accessible preprocessor for blind evaluation, we standardize on Farasa preprocessing for all subsequent experiments (full results in Appendix A.4).

Table 1 reports sentence-level results on the BAREC 2025 test set. +CP improves QWK over each baseline while reducing exact Acc, and increases ±1Acc. The strongest single model is Focal+CP (QWK 84.4; +2.6 over Focal); CE+CP and EMD+CP gain +1.6 and +1.1 QWK, respectively. The Avg and Most Common ensembles also improve QWK (to 84.9 and 84.6) and reduce Dist (down to 1.01). To quantify headroom if a user could reliably choose from the CP set, we add a non-deployable Oracle: it selects the gold label whenever it lies in the CP set, otherwise falls back to Focal+CP. This upper bound reaches QWK 95.3 and Acc 94.8, closely tracking the target coverage (α=0.10), and illustrates the potential of human-in-the-loop use of CP sets. Results on the blind test set (Table 2) validate the robustness of our approach. The ensemble averaging method achieves the highest performance at 85.7 QWK, while individual CP-enhanced models reach competitive scores of 84.3 (CE), 84.6 (EMD), and 85.3 (Focal). The regression baseline achieves 85.41 QWK, demonstrating strong performance of the regression formulation without post-processing. The consistent pattern of QWK improvements across different loss functions and evaluation sets demonstrates the generalizability of our conformal prediction approach.

5 Discussion

We analyze our conformal prediction approach with the focal loss model and APS at $\alpha = 0.1$, the best-performing setting on the dev-tune split.

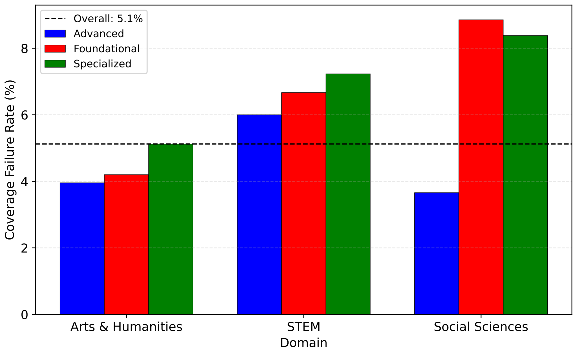


Figure 1: Coverage failure rates by domain and text class. Each domain shows three grouped bars representing Advanced, Foundational, and Specialized text classes. The dashed line shows the overall failure rate (5.12%).

The analysis highlights two aspects: (1) coverage reliability and failure patterns, (2) error redistribution underlying improvements in ordinal metrics.

5.1 CP Coverage Analysis

Using $\alpha = 0.1$ targeting 90% coverage, we report 94.88% empirical coverage with an average set size of 5 levels, a substantial reduction from the full 19-class space. This means that in nearly 95% of Arabic texts, the correct readability level appears in a compact, interpretable set. The remaining 5.12% coverage failures show systematic domain variation: 4.3% for Arts & Humanities (70/1,625), 6.1% for STEM (10/163), and 7.1% for Social Sciences (38/535). We define failure rate as the proportion of cases where the true label falls outside the conformal prediction set. Figure 1 reveals that failures are not uniformly distributed across text types. Social Sciences exhibits the highest rates, particularly for Foundational and Specialized texts (8-9% failure rates), while Arts & Humanities remains close to the overall rate. STEM shows elevated failure rates (6-7%) across all text classes. This variation suggests that domain-adaptive calibration strategies could improve coverage reliability for challenging text types. Additional coverage diagnostics are provided in Appendix A.3.

5.2 Why QWK improves despite lower exact accuracy

QWK increases because many large errors shrink while only a smaller set of perfect predictions become near misses. On the dev–tune split, CP turned 362 perfect predictions into errors (15.6%), and 86.7% of these new errors were only ±1 level.

Model Variant	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**	**Acc^{3}**	**Acc^{5}**	**Acc^{7}**
CE (Baseline)	82.6	**55.5**	71.6	1.04	79.8	71.4	**65.4**
CE + CP	84.3	50.3	72.9	1.03	**80.1**	70.1	63.8
EMD (Baseline)	82.8	54.4	71.4	1.04	79.7	**71.5**	64.6
EMD + CP	83.9	49.4	73.4	1.04	79.7	70.4	63.3
Focal (Baseline)	81.8	55.4	71.7	1.07	79.7	71.4	65.3
Focal + CP	84.4	42.7	**74.5**	1.08	78.0	67.9	61.0
Regression (Baseline)	83.8	42.0	73.2	1.12	78.0	67.3	59.8
Average	**84.9**	47.3	74.0	1.03	79.8	69.6	63.0
Most Common	84.6	49.6	74.4	**1.01**	**80.1**	70.9	64.4
Oracle Decoder	95.3	94.8	95.3	0.20	96.4	95.6	95.3

Table 1: BAREC test, sentence-level. “Baseline” = fine-tuned point decoder. “+CP” = conformal prediction (α=0.10) with our QWK-aligned mean-in-set decoder; applied to CE/EMD/Focal only (Regression is point-only). “Oracle” = upper bound that selects the gold label if it lies in the CP set; otherwise falls back to Focal+CP. All results use Farasa preprocessing.

Model Variant	**QWK**
CE (Baseline)	82.6
CE + CP	84.3
EMD (Baseline)	-
EMD + CP	84.6
Focal (Baseline)	-
Focal + CP	85.3
Regression (Baseline)	85.4
Average	**85.7**
Most Common	84.8
Document-level (Max over sentences)	73.3

Table 2: Blind test set QWK results. Missing baseline values (–) indicate models not submitted without CP enhancement. Document-level results use the maximum predicted sentence-level difficulty per document.

At the same time, 397 originally incorrect predictions improved (17.1%): 80.6% shrank by 1 level, 14.7% by 2, 3.1% by 3, and 1.6% by 4. Since QWK penalizes errors by the *squared* distance, shrinking many large mistakes yields big gains (e.g., reducing a 4-level error to 1 cuts the penalty from 16 to 1).

6 Conclusions and Future Work

We presented a simple, model-agnostic post-processing method for Arabic readability assessment that combines conformal prediction with expected value decoding. Applied to the BAREC Shared Task 2025, our approach achieved consistent QWK gains of 1-3 points across multiple base models. In the strict track, our submission achieves QWK scores of 84.9% (test) and 85.7% (blind test) for sentence level, and 73.3% for document level. Beyond leaderboard gains, the method produces compact prediction sets with statistical coverage guarantees, offering both improved accuracy and interpretable outputs for human-in-the-loop use.

Future work could extend this approach in several ways. Mondrian conformal prediction could calibrate separately for different text types or complexity ranges, potentially reducing coverage failures in difficult cases. Multi-granularity training using the BAREC mappings (3-, 5-, and 7-level schemes) may improve generalization across difficulty levels. Finally, rule-based or heuristic decoding strategies informed by the official annotation guidelines (Habash et al., 2025) could refine label selection from CP sets by leveraging linguistic cues and common annotation patterns.

7 Limitations

While our approach improves QWK and reduces high-penalty errors, several limitations remain. Most error reductions occur within medium difficulty ranges, leaving large-gap errors at higher levels (e.g., 15–19) largely unresolved. The effectiveness of our approach depends on the base model’s calibration: overconfident but incorrect probability estimates can lead to suboptimal conformal sets, and renormalization may not fully correct such biases. Finally, our CP implementation yields slightly conservative coverage (94% vs. 90% target), suggesting room for tighter calibration or adaptive thresholding.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for arabic. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations*, pages 11–16.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. 2020. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*.

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.

Yassine Benajiba and Paolo Rosso. 2008. Arabic named entity recognition using conditional random fields. In *Proc. of Workshop on HLT & NLP within the Arabic World, LREC*, volume 8, pages 143–153.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Kevyn Collins-Thompson and Jamie Callan. 2004. Information retrieval for language tutoring: An overview of the reap project. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 544–545.

Khalid Elmadani, Nizar Habash, and Hanada Taha. 2025a. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025b. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Nizar Habash, Hanada Taha-Thomure, Khalid Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Le Hou, Chen-Ping Yu, and Dimitris Samaras. 2017. Squared earth movers distance loss for training deep neural networks on ordered-classes. In *NIPS workshop*, volume 5.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

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.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. 2002. Inductive confidence machines for regression. In *European conference on machine learning*, pages 345–356. Springer.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. 2020. Classification with valid and adaptive coverage. *Advances in neural information processing systems*, 33:3581–3591.

Hind Saddiki, Nizar Habash, Violetta Cavalli-Sforza, and Muhamed Al-Khalil. 2018. Feature optimization for predicting readability of arabic l1 and l2. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 20–29.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Springer.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.

A Appendix A

A.1 Nonconformity Scores

In conformal prediction, a *nonconformity score* $s(x, y)$ quantifies how atypical a candidate label y is for an instance x given the model's output distribution $p(y \mid x)$. We evaluate three standard multiclass scoring functions:

Naïve (Inverse Probability). The simplest approach uses the complement of the predicted probability:

$$s_{\text{naive}}(x, y) = 1 - p(y \mid x) \tag{3}$$

This yields smaller scores for high-probability labels, producing larger prediction sets for low-confidence predictions.

Adaptive Prediction Sets (APS) (Romano et al., 2020). Let $\pi_1, \pi_2, \ldots, \pi_K$ denote the classes sorted in descending order of their probabilities $p(\pi_1 \mid x) \geq p(\pi_2 \mid x) \geq \cdots \geq p(\pi_K \mid x)$. For a given label y, let $r(y)$ be its rank in this sorted order. The APS score is the cumulative probability mass up to and including label y:

$$s_{\text{aps}}(x, y) = \sum_{j=1}^{r(y)} p(\pi_j \mid x) \tag{4}$$

Regularized Adaptive Prediction Sets (RAPS) (Angelopoulos et al., 2020). RAPS extends APS by adding a linear rank-based penalty:

$$s_{\text{raps}}(x, y) = \sum_{j=1}^{r(y)} p(\pi_j \mid x) + \lambda \cdot r(y) \tag{5}$$

where $\lambda \geq 0$ is the regularization parameter controlling the size-coverage trade-off. In this work, we set $\lambda = 0.01$.

A.2 Experimental Setup

All experiments were conducted on a single NVIDIA A100 GPU using Google Colab Pro. Training was performed for 6 epochs with a batch size of 64, a learning rate of 5×10^{-5}, and the Adam optimizer. The best checkpoint was selected based on development set performance measured by Quadratic Weighted Kappa (QWK).

A.3 CP Coverage Plots

To better understand the behavior of our conformal prediction variants, we provide supplementary plots analyzing performance, coverage calibration, and set size trends across different miscoverage rates α. In Figure 2, we show the relationship between miscoverage rate α and Quadratic Weighted Kappa (QWK) for three conformal prediction methods on the dev-tune set. APS and RAPS maintain stable QWK across all α values, consistently outperforming the baseline. The naïve method degrades sharply beyond $\alpha > 0.2$, indicating poor robustness when allowing larger miscoverage. In Figure 3, we plot the actual coverage against the target coverage for Naïve, APS, and RAPS methods. All methods achieve coverage above the target across the range, indicating slight conservativeness. This effect is most pronounced for APS, which consistently overshoots the target coverage. Such conservative calibration ensures statistical validity but may produce larger prediction sets than necessary, potentially impacting their interpretability. Finally, figure 4 shows the relationship between the miscoverage rate α and the average prediction set size for the three nonconformity scoring methods. For α, APS and RAPS yield larger sets than the naïve method, with APS producing the widest sets.

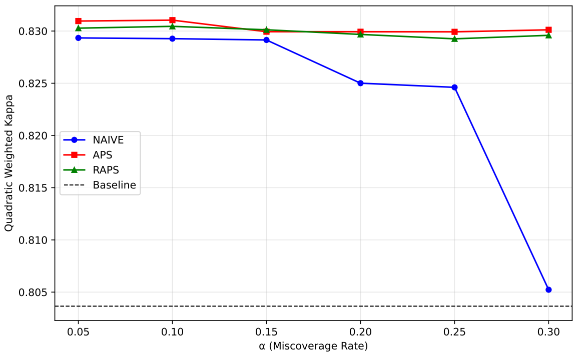


Figure 2: Quadratic Weighted Kappa performance vs. miscoverage rate (α) for three conformal prediction scoring methods on the dev-tune split. The dashed line represents baseline performance without conformal prediction.

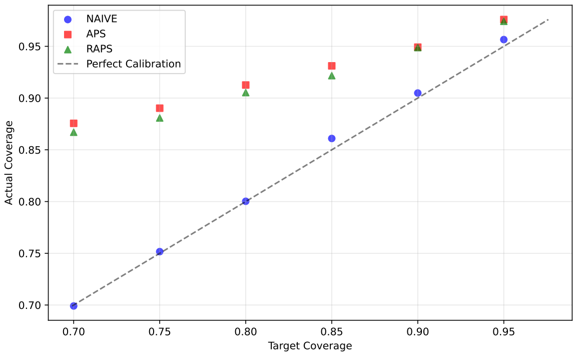


Figure 3: Coverage calibration quality showing actual vs. target coverage rates. The dashed line represents perfect calibration where actual coverage equals target coverage.

A.4 Preprocessing & Loss Ablations

A.5 Dev Data Split

Loss	**Input**	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**
CE	Word	77.6	53.4	68.2	1.24
CE	Lex	76.4	49.0	66.1	1.32
CE	D3Lex	79.8	53.0	68.3	1.19
CE	D3Tok	81.4	53.3	**70.9**	1.14
CE	Farasa	**80.2**	**55.5**	70.6	**1.13**
EMD	Word	78.2	52.0	67.3	1.24
EMD	Lex	79.5	48.8	66.8	1.24
EMD	D3Lex	80.4	52.2	68.3	1.18
EMD	D3Tok	81.2	53.1	69.8	1.13
EMD	Farasa	**81.4**	**54.8**	**71.0**	**1.10**
Regression	Word	79.3	38.5	69.4	1.30
Regression	Lex	80.9	35.8	69.2	1.31
Regression	D3Lex	82.3	38.7	70.7	1.26
Regression	D3Tok	82.4	40.7	71.5	1.20
Regression	Farasa	**82.9**	**43.3**	**72.5**	**1.15**
Focal	Word	77.6	52.6	67.6	1.25
Focal	Lex	77.9	49.4	67.0	1.27
Focal	D3Lex	80.0	53.4	69.1	1.18
Focal	D3Tok	**80.5**	56.0	**71.1**	**1.12**
Focal	Farasa	80.4	**56.1**	71.0	**1.12**

Table 3: AraBERTv2 results on the BAREC Development set across different loss functions and input representations.

Loss	**Input**	**QWK**	**Acc^{19}**	**±1 Acc^{19}**	**Dist**
CE	Word	79.2	54.0	68.6	1.17
CE	Lex	78.4	49.7	66.9	1.23
CE	D3Lex	80.6	53.2	68.1	1.14
CE	D3Tok	81.9	52.8	70.9	1.10
CE	Farasa	**82.6**	**55.5**	**71.6**	**1.04**
EMD	Word	80.7	53.3	68.9	1.13
EMD	Lex	80.6	49.6	67.0	1.18
EMD	D3Lex	81.3	53.3	69.6	1.11
EMD	D3Tok	81.7	52.7	69.3	1.10
EMD	Farasa	**82.8**	**54.5**	**71.4**	**1.04**
Regression	Word	81.4	38.8	70.4	1.23
Regression	Lex	81.4	35.5	70.1	1.26
Regression	D3Lex	82.8	39.2	70.9	1.18
Regression	D3Tok	83.1	40.7	72.2	1.15
Regression	Farasa	**83.8**	**42.0**	**73.2**	**1.11**
Focal	Word	79.9	53.9	69.4	1.14
Focal	Lex	79.5	50.6	67.7	1.19
Focal	D3Lex	80.9	53.1	69.6	1.13
Focal	D3Tok	**82.2**	55.2	71.2	**1.06**
Focal	Farasa	81.8	**55.4**	**71.7**	1.07

Table 4: AraBERTv2 results on the BAREC Test set across different loss functions and input representations.

Class	Original	%	Dev-Cal	Dev-Tune	Split Ratio
1	44	0.6	32	12	73:27
2	68	0.9	49	19	72:28
3	182	2.5	126	56	69:31
4	78	1.1	55	23	71:29
5	417	5.7	284	133	68:32
6	189	2.6	130	59	69:31
7	701	9.6	476	225	68:32
8	613	8.4	417	196	68:32
9	236	3.2	162	74	69:31
10	1012	13.8	686	326	68:32
11	409	5.6	279	130	68:32
12	1491	20.4	1010	481	68:32
13	349	4.8	239	110	68:32
14	1072	14.7	727	345	68:32
15	258	3.5	177	81	69:31
16	114	1.6	80	34	70:30
17	49	0.7	36	13	73:27
18	13	0.2	10	3	77:23
19	15	0.2	12	3	80:20
Total	**7310**	**100.0**	**4981**	**2329**	**68:32**

Table 5: Development set stratified split into calibration (Dev-Cal) and tuning (Dev-Tune) subsets.

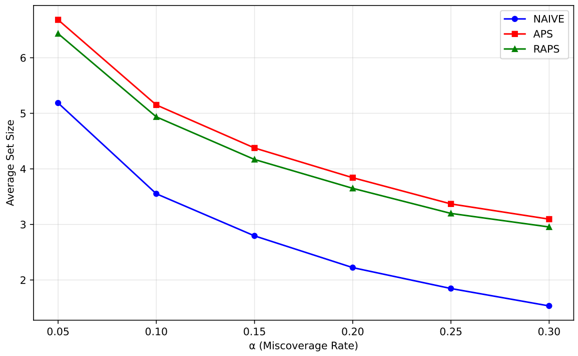


Figure 4: Average prediction set sizes across miscoverage rate (α) for the three conformal prediction scoring methods.

AMAR at BAREC Shared Task 2025: Arabic Meta-learner for Assessing Readability

Mostafa Saeed,[1] Rana Waly,[2] Abdelaziz Ashraf Hussein[3]
[1]New York University Abu Dhabi, UAE
[2]Digital Egypt for Investment Co., Cairo, Egypt
[3]Graduate School of Science and Engineering, Ozyegin University, 34794 İstanbul, Türkiye
mms10094@nyu.edu, rana.reda@defi.com.eg, abdelaziz.hussein@ozu.edu.tr

Abstract

Navigating the complexities of Arabic readability prediction requires addressing the language's rich morphology and structural diversity. In the BAREC Shared Task 2025, we participated in all tracks using a stacked ensemble meta learning framework. Our approach combined seven fine-tuned transformer, whose outputs fed into a meta classifier trained on multiple features, including individual predictions, their average, and the average top prediction probabilities. On the blind test set, our ensemble achieved a Quadratic Weighted Kappa (QWK) of 86.4%, demonstrating the effectiveness of integrating diverse transformer encoders for fine grained Arabic readability classification and the potential of meta learning in morphologically rich contexts.

1 Introduction

Arabic readability prediction assesses how difficult a text is for its intended audience, supporting applications such as text simplification (Fang et al., 2025), adaptive learning (Fitrianto et al., 2024), and automated grading (Qwaider et al., 2025). In Arabic, the task is particularly challenging due to the language's morphological richness and wide dialectal variation, and it also plays a crucial role in promoting equitable access to information for readers of varying proficiency levels.

The Balanced Arabic Readability Evaluation Corpus (BAREC) dataset (Elmadani et al., 2025a) heightens this complexity by covering multiple genres from news, literature, educational content, children poems, social media and more other genres that was discussed by them. This diversity introduces significant lexical, syntactic, and stylistic variation, requiring models to capture cues from orthographic patterns to higher level semantics.

In our effort to contribute to this evolving field, we participated in the BAREC Shared Task 2025 (Elmadani et al., 2025b), which focuses on sentence and document level Arabic readability prediction across 19 distinct difficulty levels. The competition comprises three tracks: a Strict track, where only BAREC data is permitted for training; a Constrained track, where the BAREC dataset, SAMER corpus (Alhafni et al., 2024), and SAMER lexicon (Al Khalil et al., 2020) are available; and an Open track, where any external resources may be used.

Our main objective was to assess whether a stacked meta learning system could achieve competitive performance by leveraging the strengths of several fine-tuned transformer models. In this framework, seven transformer based language models served as base predictors, followed by a meta classifier trained on multiple features details of which will be discussed later on to predict the final readability level. We extended the system in the constrained track by incorporating lexical features extracted from the SAMER lexicon which is an arabic readability resource that assigns difficulty levels to individual words, making it possible to estimate text complexity based on its lexical content. In the open track, we also explored a prompt based zero shot approach with GPT 4.1, by feeding the model with a structured annotation guidelines to guide and refine its predictions.

Our stacked meta learning system achieved 2nd place in Track 1 (sentence level) and 2nd in Track 2 (sentence level), but only 7th in document level Track 1 and was not tested in Track 2 due to poor performance. This indicates its strength at the sentence level but limited effectiveness for documents, partly due to a trade off between QWK and accuracy. Employing the LLM for human like annotation was also ineffective.

The paper is organized as follows: §2 reviews related work, §3 presents the dataset, §4 the methodology, §5 the results, §6 the discussion, and §8 the conclusion and future work.

2 Related Work

Zalmout et al. (2016) showed that early automatic readability assessment relied on traditional formulas like Flesch Reading Ease (Flesch and Gould, 1949), Flesch Kincaid (Kincaid et al., 1975), and Dale Chall (Dale and Chall, 1948), focusing on surface features such as sentence and word length and vocabulary familiarity. They later extended this by incorporating lexical and syntactic features into SVMs.

For Arabic, Saddiki et al. (2018) conducted the most extensive study, employing a wide range of lexical and syntactic features for L1 and L2 tasks. Their results demonstrated that leveraging L1 features can improve L2 readability prediction, highlighting the benefits of cross task feature sharing.

Ambati et al. (2016) compared syntactic features from incremental CCG and non-incremental phrase parsers, showing that incremental parsing enhanced both accuracy and speed, with further improvements from adding psycholinguistic features. Similarly, Deutsch et al. (2020) found that neural models, ranging from SVMs to BERT, can match or outperform feature augmented systems when trained on sufficient data, suggesting that deep models already capture key readability indicators.

Liberato et al. (2024) introduced a multi model framework for Arabic word level readability (Hazim et al., 2022) and fragment level readability, combining lexicon, frequency, statistical, and transformer based models, and demonstrated that cascaded and aggregation strategies yield stronger results. Recent research further explores deep learning approaches (Lee and Vajjala, 2022; Imperial and Kochmar, 2023) and the use of large language models (LLMs) (Naous et al., 2024; Huang et al., 2024; Marulli et al., 2024), leveraging their advanced language understanding to predict and analyze readability with greater nuance.

Building on prior work, we participated in all three tracks of the shared task, Track 1 (sentence and document level), Track 2 (sentence level), and Track 3 (sentence level) exploring two main directions: **(1)** integrating machine learning models with fine-tuned models to leverage the strengths of both through a stacked meta classifier in Track 1&2, and **(2)** evaluating the capacity of LLMs in Track 3 to emulate human annotation through systematic prompt engineering.

3 Data

3.1 BAREC Dataset

The Balanced Arabic Readability Evaluation Corpus (BAREC) is a large scale dataset for Arabic readability assessment, containing 69,441 manually annotated sentences (over one million words) across 19 readability levels, ranging from kindergarten to postgraduate. It is designed to balance genre, topic, and audience coverage, providing a rich resource for evaluating Arabic text complexity.

The dataset is divided into four subsets: training, validation, and public test splits, which are provided during the development phase, and a private test set, which is used to evaluate the final systems after the development phase concludes.

We conducted thorough evaluations using the validation and public test sets, followed by a final assessment of the system on the blind test set provided for the shared task.

	# Docs	# Sentences	# Words
Train	1,518	54,845	832,743
Dev	194	7,310	101,364
Public Test	210	7,286	105,264
Blind Test	100	3,420	53,052

Table 1: Dataset statistics for the training, validation, public test, and blind test sets.

3.2 SAMER Lexicon

We present the SAMER Lexicon, a 40K lemma leveled readability resource for Arabic. The lexicon comprises 40,000 lemma and part of speech pairs, each annotated with one of five readability levels. This resource offers a standardized reference for assessing lexical difficulty, enabling its integration into a wide range of readability prediction and educational technology applications.

4 Methodology

4.1 Overview

In this paper, we present our submissions for the three tracks of the BAREC shared task. For Tracks 1 and 2, we followed the recommendation from the BAREC main paper, which suggested framing the task as a regression problem to achieve higher QWK scores. Building on this, we fine-tuned multiple transformers and then trained a stacked meta classifier ML model to predict the final readability level based on their outputs. In contrast, Track 3 adopts a fundamentally different approach: we experimented with LLMs, specifically leveraging the ChatGPT 4.1, to generate predictions directly through prompt based inference.

4.2 Track 1 & Track 2: Stacked Meta Classifier Approach

The preprocessing stage involved removing both kashida and all diacritics from the text. We then fine-tuned several transformer based models, where the input was the complete sentence and the target label was the readability level of that sentence. The outputs from the fine-tuned models were used as inputs to a stacked meta classifier with three feature types: **(1)** raw predictions from each model, **(2)** their average, and **(3)** the average of the top prediction probabilities. We tested these features individually and in combination. For efficiency in deployment, the meta classifier was implemented as a lightweight ML model (classifier or regressor) operating on model predictions rather than raw text.

We experimented with using these features individually and in combination, presenting only the best results achieved through the combination of all three features. To ensure computational efficiency in the final deployment stage, we implemented the meta classifier as a lightweight machine learning model (either a classifier or a regressor) that operates on the predictions of the fine-tuned models.

Added Lexical Features for Track 2 While Track 2 used the same pipeline, we augmented the meta classifier input with three lexical features from the SAMER lexicon. Each word was lemmatized using the CAMeL Tools MSA disambiguator (Obeid et al., 2020), then matched to the SAMER lexicon; if not found, the closest match was selected via edit distance. For each sentence, we calculated **(1)** the most frequent, **(2)** the maximum, and **(3)** the average SAMER level, which were concatenated with the existing meta classifier features to improve prediction accuracy.

We fine-tuned several transformer based models, including AraBERTv02 (Antoun et al., 2020), AraBERTv2 (Antoun et al., 2020), MARBERTv2 (Abdul-Mageed et al., 2021), bert base arabic camelbert msa (Inoue et al., 2021), XLM RoBERTa large (Conneau et al., 2019), bert qarib (Abdelali et al., 2021) and NuSentiment multilingual (Wang et al., 2024), following the regression based setup described earlier.

For the document level setting, we applied the same process at the sentence level, then assigned each document the maximum readability level predicted for any sentence it contained.

4.3 Track 3: LLM Based Approach

In this track, our goal was to emulate human annotation using a powerful LLM by embedding the full Arabic annotation guidelines (Habash et al., 2025) into the prompt. These guidelines define the evaluator's role, describe the 19 readability levels across six linguistic dimensions, and provide examples, constraints, and ACTFL aligned progression from simplest to most complex. By embedding these criteria in the prompt, we guided the LLM to produce annotations consistent with human judgments, enhanced through prompt engineering techniques such as role specification, task definition, criteria conditioning, and strict output formatting.

5 Results

5.1 Sentence Level

5.1.1 Track 1 & 2

We evaluated the performance of the fine-tuned models individually as well as within the meta learning framework. As shown in table 2, **CAMeLBERT-MSA** achieved the highest performance among all base models, with a QWK of 82.8% on the development set and 83.8% on the public test set.

Model	Dev-QWK	Dev-Acc	Test-QWK	Test-Acc
Arabertv2	77.9%	27.0%	78.8%	27.2%
Arabertv02	80.9%	29.9%	81.9%	29.3%
MArabertv2	81.4%	28.1%	82.1%	28.2%
camel_bert_msa	**82.8%**	36.7%	**83.8%**	36.5%
XLM-ROBERTA	80.6%	**38.5%**	81.8%	**39.3%**
bert_qarib	79.9%	26.6%	81.3%	26.0%
Nu_sent	81.1%	27.8%	82.1%	27.9%

Table 2: Performance of base models on the dev and public test sets (Track 1, sentence-level prediction) for QWK and accuracy.

For the ensemble setting, we conducted an extensive series of experiments using a wide range of machine learning classifiers and regressors. As shown in Table 3, the Naïve Bayes models both Gaussian and Categorical consistently yielded the best results. This result was observed when training on the individual predictions of the seven fine-tuned models, and further improved when incorporating the average score across models. We explored all possible combinations of model predictions, and the best performance was achieved when using the predictions from all seven models together. Performance increased even more when we additionally included the average of the top predicted probabilities for each instance.

Model	Dev-QWK	Dev-Acc	Test-QWK	Test-Acc
Logistic Regression	81.9%	**45.1%**	82.8%	**44.5%**
Linear Regression	82.6%	37.9%	83.8%	39.0%
Random Forest Classifier	81.0%	41.7%	81.5%	41.4%
Random Forest Regressor	81.8%	39.5%	82.7%	39.8%
GaussianNB	**83.9%**	39.2%	**84.9%**	38.1%
CategoricalNB	83.7%	38.1%	**84.9%**	37.6%
Bagging Classifier	80.6%	42.1%	81.1%	41.6%
Bagging Regressor	81.6%	39.9%	82.5%	39.4%

Table 3: Performance of the meta classifier on the dev and public test sets

For track 2, The same ensemble configuration was then applied with the addition of the previously discussed features. As shown in Table 4, this led to only a marginal improvement on the overall performance.

Model	Dev-QWK	Dev-Acc	Test-QWK	Test-Acc
Logistic Regression	81.8%	**45.1%**	82.8%	**44.4%**
Linear Regression	82.7%	38.0%	83.8%	39.1%
Random Forest Classifier	81.7%	44.9%	81.5%	44.7%
Random Forest Regressor	82.0%	40.3%	82.7%	40.6%
GaussianNB	**83.9%**	38.9%	**84.9%**	37.7%
CategoricalNB	83.7%	38.1%	**84.9%**	37.7%
Bagging Classifier	80.9%	44.3%	81.1%	43.6%
Bagging Regressor	81.5%	39.8%	82.5%	39.9%

Table 4: Performance of the meta classifier on the dev and public test sets.

We selected the CategoricalNB model due to its outstanding performance on the public test set and applied it to the blind test. The results are presented in Table 5.

Track	Model	QWK	Acc
Track 1	CategoricalNB	86.4%	39.7%
Track 2	CategoricalNB	86.4%	39.9%

Table 5: Overall performance on Track 1 and Track 2 Blind test.

5.1.2 Track 3

For the LLM trial, even after extensive prompt engineering and providing the ChatGPT 4.1 API with the full BAREC guidelines, performance was poor, achieving only 40.7% QWK on the dev set when predicitng on the sentence level.

5.2 Document Level

For the document level assessment, we applied the previously described approach on Track1; however, it yielded suboptimal results with 69.6% QWK and 34% accuracy. The reasons for this underperformance are examined in detail in the Discussion section. Since the results were unsatisfactory on Track1, we did not extend this approach to Track 2.

6 Discussion

The results show that the stacked meta learner classifier has a strong positive impact compared to individual fine-tuned models, with CategoricalNB achieving slightly better performance than GaussianNB for sentence level predictions. However, this approach did not transfer well to the document level, where higher accuracy is crucial. Regression based models, while yielding high QWK, tend to have lower accuracy, which limits their effectiveness for document level prediction.

Adding the lexical features produced only a marginal improvement of 0.2% in accuracy, indicating limited impact.

For Track 3, using GPT-4.1 with the provided guidelines and a few prompt engineering techniques performed poorly, failing to effectively mimic the human annotation process.

7 Error Analysis

As shown in appendix figures 3 and 4 , the model excels on Classes 10, 8, and 7 but struggles with 6, 1, and 5, often confusing them with neighboring levels. Mid levels are more distinct, while lower levels exhibit significant overlap.

8 Conclusion & Future Work

This shared task provided a valuable opportunity to advance Arabic readability prediction by comparing diverse modeling strategies across three tracks. Our results highlight the effectiveness of a stacked meta learner, which consistently outperformed individual fine-tuned transformer models, with CategoricalNB delivering the best sentence level results. However, the approach proved less effective for document level prediction, where the accuracy QWK trade off in the fine-tuned models.

These findings emphasize the need for models that balance both accuracy and QWK for document level prediction, as well as more impactful feature integration strategies. Future directions include exploring hybrid architectures, leveraging contextual lexical embeddings, and developing advanced prompting or fine-tuning methods for LLMs to better align outputs with human judgments.

Ethics Statement

Although certain authors maintain institutional affiliations with entities linked to the shared-task organizers, the organizers themselves did not contribute to the ideation, construction, or testing of our systems. The entirety of our work was conducted using only those resources that were openly and uniformly distributed to the participant community, with no selective access or special guidance granted.

Acknowledgments

We acknowledge the contribution of the High Performance Computing Center at NYU Abu Dhabi in providing the necessary support for this research.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

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.

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Bharat Ram Ambati, Siva Reddy, and Mark Steedman. 2016. Assessing relative sentence complexity using an incremental CCG parser. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1057, San Diego, California. Association for Computational Linguistics.

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17, Seattle, WA, USA → Online. Association for Computational Linguistics.

Khalid Elmadani, Nizar Habash, and Hanada Taha. 2025a. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025b. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Dengzhao Fang, Jipeng Qiang, Yi Zhu, Yunhao Yuan, Wei Li, and Yan Liu. 2025. Progressive document-level text simplification via large language models. *arXiv preprint arXiv:2501.03857*.

Ibnu Fitrianto, Cahya Edi Setyawan, and Malikus Saleh. 2024. Utilizing artificial intelligence for personalized arabic language learning plans. *International Journal of Post Axial: Futuristic Teaching and Learning*, pages 30–40.

Rudolf Franz Flesch and Alan J Gould. 1949. The art of readable writing. *(No Title)*.

Nizar Habash, Hanada Taha-Thomure, Khalid Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Chieh-Yang Huang, Jing Wei, and Ting-Hao Kenneth Huang. 2024. Generating educational materials with different levels of readability using llms. In *Proceedings of the Third Workshop on Intelligent and Interactive Writing Assistants*, pages 16–22.

Joseph Marvin Imperial and Ekaterina Kochmar. 2023. Automatic readability assessment for closely related languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5371–5386, Toronto, Canada. 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*, Kyiv, Ukraine (Online). Association for Computational Linguistics.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report.

Justin Lee and Sowmya Vajjala. 2022. A neural pairwise ranking model for readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3802–3813, Dublin, Ireland. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Fiammetta Marulli, Lelio Campanile, Maria Stella de Biase, Stefano Marrone, Laura Verde, and Marianna Bifulco. 2024. Understanding readability of large language models output: an empirical analysis. *Procedia Computer Science*, 246:5273–5282.

Tarek Naous, Michael J Ryan, Anton Lavrouk, Mohit Chandra, and Wei Xu. 2024. ReadMe++: Benchmarking multilingual language models for multidomain readability assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12230–12266, Miami, Florida, USA. 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.

Chatrine Qwaider, Bashar Alhafni, Kirill Chirkunov, Nizar Habash, and Ted Briscoe. 2025. Enhancing Arabic automated essay scoring with synthetic data and error injection. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 549–563, Vienna, Austria. Association for Computational Linguistics.

Hind Saddiki, Nizar Habash, Violetta Cavalli-Sforza, and Muhamed Al Khalil. 2018. Feature optimization for predicting readability of Arabic L1 and L2. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 20–29, Melbourne, Australia. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Nasser Zalmout, Hind Saddiki, and Nizar Habash. 2016. Analysis of foreign language teaching methods: An automatic readability approach. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 122–130, Osaka, Japan. The COLING 2016 Organizing Committee.

A Full Evaluation Metrices per approach

Tables 6, 7 and 8 present the extended results, including the four metrics reported in the BAREC paper. As noted earlier, both CategoricalNB and GaussianNB consistently achieve high QWK scores relative to the other models. However, on the blind test set, CategoricalNB consistently outperforms GaussianNB.

Model	D-QWK	D-Acc	D-±1	D-Dist	T-QWK	T-Acc	T-±1	T-Dist
Arabertv2	77.9%	27.0%	63.7%	1.41	78.8%	27.2%	64.4%	1.36
Arabertv02	80.9%	29.9%	68.1%	1.31	81.9%	29.3%	69.2%	1.27
MArabertv2	81.4%	28.1%	66.8%	1.36	82.1%	28.2%	67.1%	1.31
camel_bert_msa	**82.8%**	36.7%	**71.5%**	**1.22**	**83.8%**	36.5%	**72.3%**	**1.17**
XLM-RoBERTa	80.6%	**38.5%**	70.6%	1.26	81.8%	**39.3%**	71.5%	1.20
bert_qarib	79.9%	26.6%	66.9%	1.37	81.3%	26.0%	68.2%	1.31
Nu_sent	81.1%	27.8%	66.6%	1.38	82.1%	27.9%	66.7%	1.33

Table 6: Extended performance of base models on the dev (D) and public test (T) sets (Track 1: Sentence Level), including QWK, accuracy, ±1 accuracy, and distribution distance.

Model	D-QWK	D-Acc	D-±1	D-Dist	T-QWK	T-Acc	T-±1	T-Dist
Logistic Regression	81.9%	45.1%	64.7%	0.36	82.8%	44.5%	65.3%	0.34
Linear Regression	82.6%	37.9%	**72.0%**	0.36	83.8%	39.0%	**72.1%**	0.37
Random Forest Classifier	81.0%	41.7%	65.5%	0.31	81.5%	41.4%	66.3%	**0.26**
Random Forest Regressor	81.8%	39.5%	68.9%	0.30	82.7%	39.8%	69.7%	0.29
GaussianNB	**83.9%**	39.2%	66.8%	**0.27**	**84.9%**	38.1%	67.1%	0.30
CategoricalNB	83.7%	38.1%	70.1%	0.34	**84.9%**	37.6%	70.3%	0.34
Bagging Classifier	80.6%	**42.1%**	65.8%	0.29	81.1%	**41.6%**	66.3%	0.30
Bagging Regressor	81.6%	39.9%	68.0%	0.29	82.5%	39.4%	68.7%	0.28

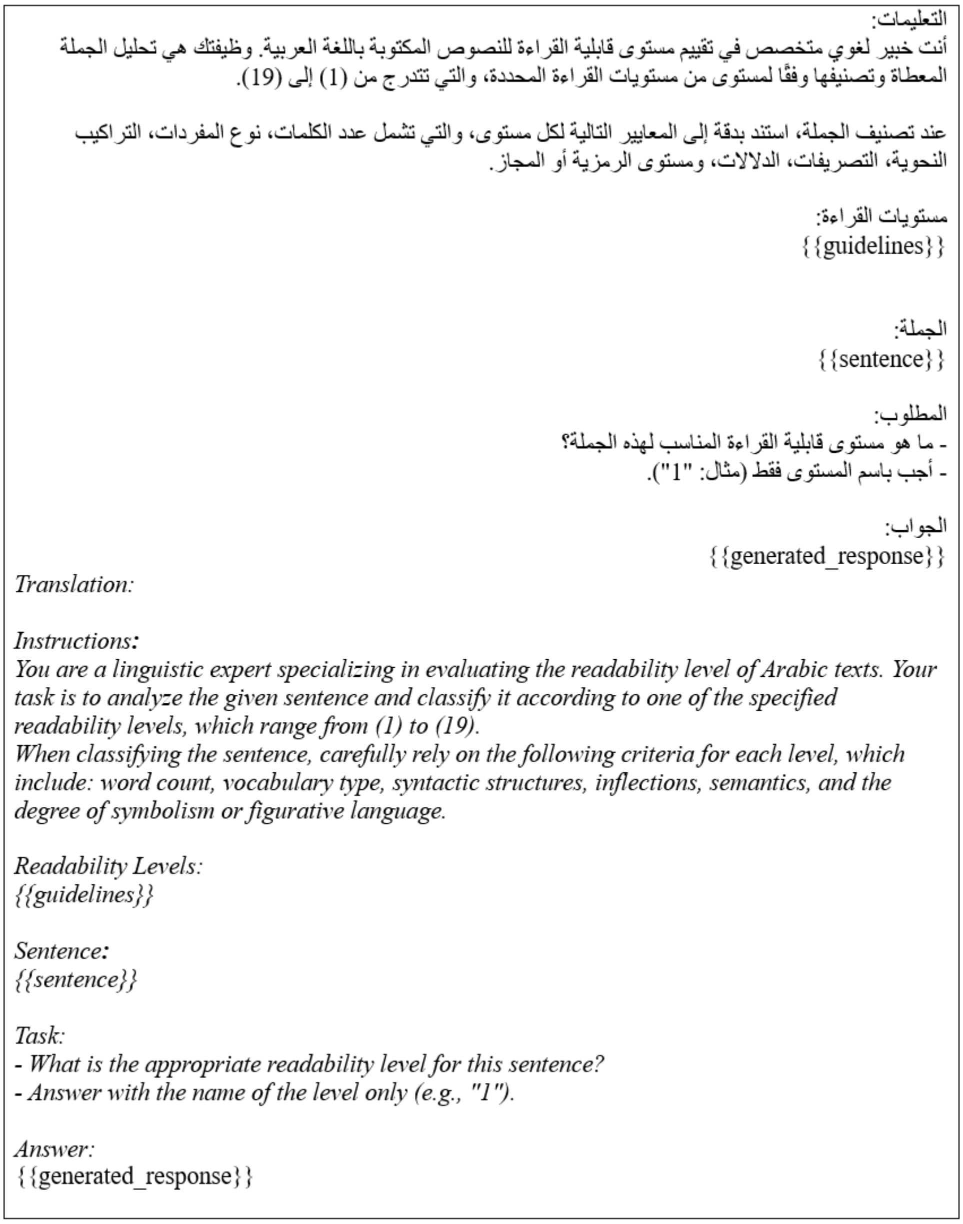
Table 7: Extended performance of ensemble models on the dev (D) and public test (T) sets (Track 1: Sentence Level), including QWK, accuracy, ±1 accuracy, and distribution distance.

Model	D-QWK	D-Acc	D-±1	D-Dist	T-QWK	T-Acc	T-±1	T-Dist
Logistic Regression	81.8%	45.1%	64.7%	0.36	82.8%	44.4%	65.3%	0.33
Linear Regression	82.7%	38.0%	**72.0%**	0.35	83.8%	39.1%	**72.0%**	0.37
Random Forest Classifier	81.7%	44.9%	66.6%	0.30	81.5%	44.7%	67.2%	**0.28**
Random Forest Regressor	82.0%	40.3%	70.1%	0.31	82.7%	40.6%	70.7%	0.30
GaussianNB	**83.9%**	38.9%	66.6%	**0.27**	**84.9%**	37.7%	67.0%	0.30
CategoricalNB	83.7%	38.1%	70.1%	0.34	**84.9%**	37.7%	70.5%	0.33
Bagging Classifier	80.9%	**44.3%**	66.4%	0.31	81.1%	**43.6%**	67.3%	0.32
Bagging Regressor	81.5%	39.8%	69.2%	0.31	82.5%	39.9%	70.1%	0.30

Table 8: Extended performance of ensemble models on the dev (D) and public test (T) sets (Track 2: Sentence Level), including QWK, accuracy, ±1 accuracy, and distributional distance.

B Prompt Details and Example

The following figure 1 illustrates the prompt used in the third track of the shared task, where we explored prompt based zero shot classification using GPT 4.1. In this setting, the model was provided with structured BAREC annotation guidelines to mimic human labeling. Figure 2 presents the guidelines extracted from Habash et al. (2025), which were embedded in the prompt to serve as a rubric or set of criteria for guiding the model in selecting the appropriate readability level.



التعليمات:
أنت خبير لغوي متخصص في تقييم مستوى قابلية القراءة للنصوص المكتوبة باللغة العربية. وظيفتك هي تحليل الجملة المعطاة وتصنيفها وفقًا لمستوى من مستويات القراءة المحددة، والتي تتدرج من (1) إلى (19).

عند تصنيف الجملة، استند بدقة إلى المعايير التالية لكل مستوى، والتي تشمل عدد الكلمات، نوع المفردات، التراكيب النحوية، التصريفات، الدلالات، ومستوى الرمزية أو المجاز.

مستويات القراءة:
{{guidelines}}

الجملة:
{{sentence}}

المطلوب:
- ما هو مستوى قابلية القراءة المناسب لهذه الجملة؟
- أجب باسم المستوى فقط (مثال: "1").

الجواب:
{{generated_response}}

Translation:

Instructions:
You are a linguistic expert specializing in evaluating the readability level of Arabic texts. Your task is to analyze the given sentence and classify it according to one of the specified readability levels, which range from (1) to (19).
When classifying the sentence, carefully rely on the following criteria for each level, which include: word count, vocabulary type, syntactic structures, inflections, semantics, and the degree of symbolism or figurative language.

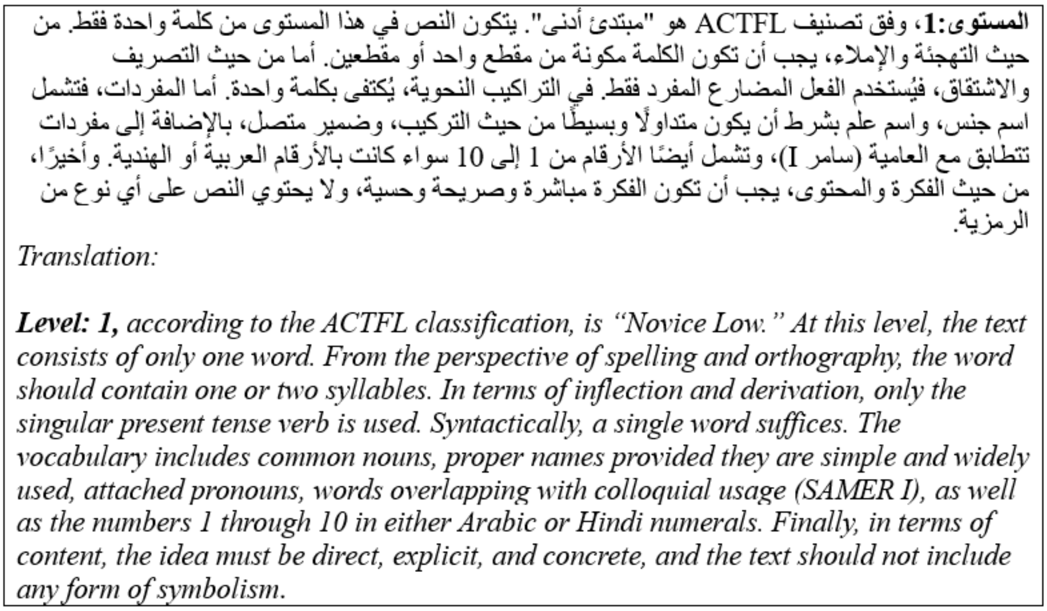
Readability Levels:
{{guidelines}}

Sentence:
{{sentence}}

Task:
- *What is the appropriate readability level for this sentence?*
- *Answer with the name of the level only (e.g., "1").*

Answer:
{{generated_response}}

Figure 1: Prompt example for the Arabic Readability Assessment



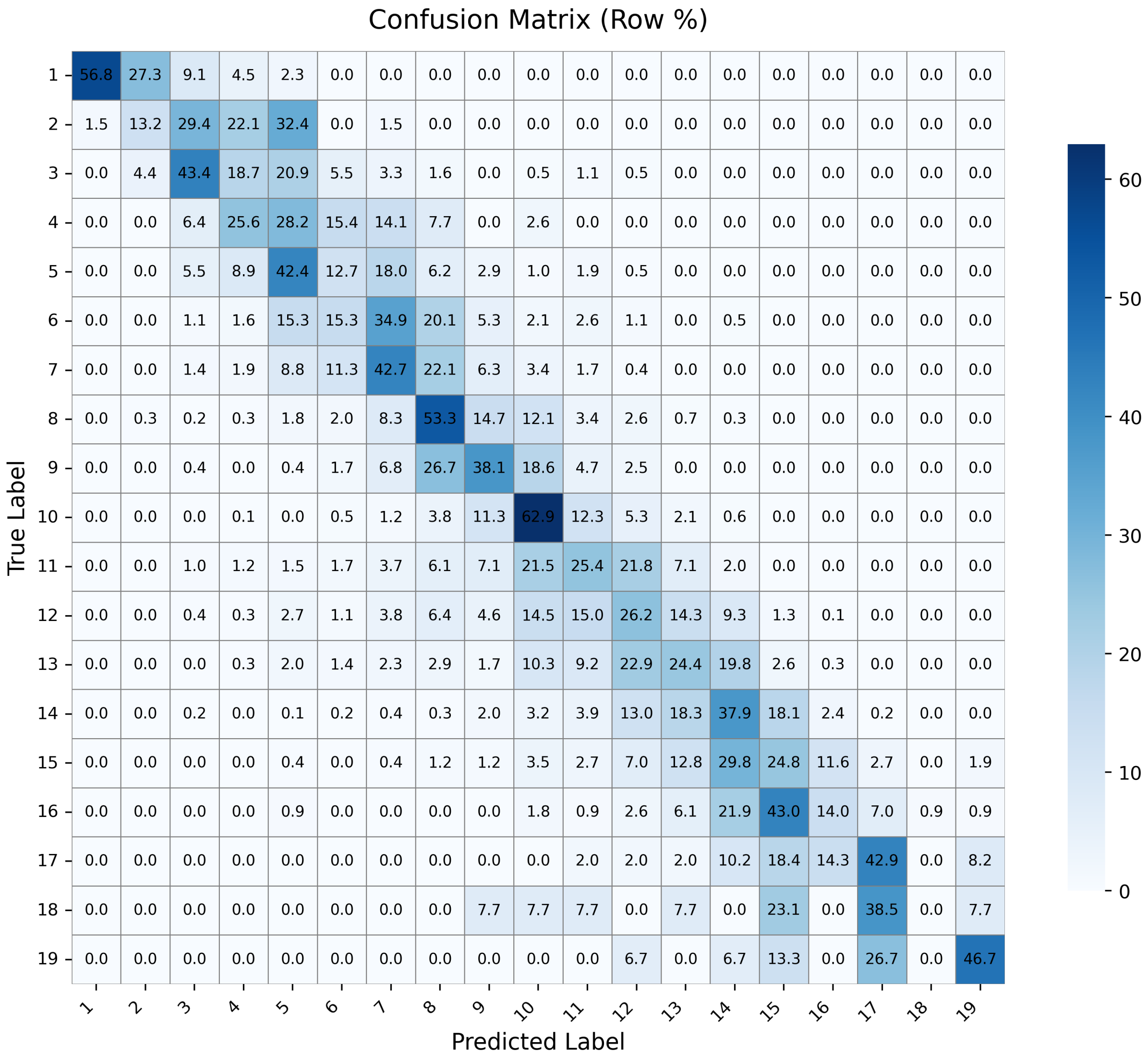
المستوى:1، وفق تصنيف ACTFL هو "مبتدئ أدنى". يتكون النص في هذا المستوى من كلمة واحدة فقط. من
حيث التهجئة والإملاء، يجب أن تكون الكلمة مكونة من مقطع واحد أو مقطعين. أما من حيث التصريف
والاشتقاق، فيُستخدم الفعل المضارع المفرد فقط. في التراكيب النحوية، يُكتفى بكلمة واحدة. أما المفردات، فتشمل
اسم جنس، واسم علم بشرط أن يكون متداولًا وبسيطًا من حيث التركيب، وضمير متصل، بالإضافة إلى مفردات
تتطابق مع العامية (سامر I)، وتشمل أيضًا الأرقام من 1 إلى 10 سواء كانت بالأرقام العربية أو الهندية. وأخيرًا،
من حيث الفكرة والمحتوى، يجب أن تكون الفكرة مباشرة وصريحة وحسية، ولا يحتوي النص على أي نوع من
الرمزية.

Translation:

Level: 1, *according to the ACTFL classification, is "Novice Low." At this level, the text consists of only one word. From the perspective of spelling and orthography, the word should contain one or two syllables. In terms of inflection and derivation, only the singular present tense verb is used. Syntactically, a single word suffices. The vocabulary includes common nouns, proper names provided they are simple and widely used, attached pronouns, words overlapping with colloquial usage (SAMER I), as well as the numbers 1 through 10 in either Arabic or Hindi numerals. Finally, in terms of content, the idea must be direct, explicit, and concrete, and the text should not include any form of symbolism.*

Figure 2: Example of the guidelines used in the prompt to differentiate between the 19 different readability levels

C Error Analysis Confusion Matrix



True \ Predicted	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	56.8	27.3	9.1	4.5	2.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1.5	13.2	29.4	22.1	32.4	0.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	4.4	43.4	18.7	20.9	5.5	3.3	1.6	0.0	0.5	1.1	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	6.4	25.6	28.2	15.4	14.1	7.7	0.0	2.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	5.5	8.9	42.4	12.7	18.0	6.2	2.9	1.0	1.9	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	1.1	1.6	15.3	15.3	34.9	20.1	5.3	2.1	2.6	1.1	0.0	0.5	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	1.4	1.9	8.8	11.3	42.7	22.1	6.3	3.4	1.7	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.3	0.2	0.3	1.8	2.0	8.3	53.3	14.7	12.1	3.4	2.6	0.7	0.3	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.4	0.0	0.4	1.7	6.8	26.7	38.1	18.6	4.7	2.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.1	0.0	0.5	1.2	3.8	11.3	62.9	12.3	5.3	2.1	0.6	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	1.0	1.2	1.5	1.7	3.7	6.1	7.1	21.5	25.4	21.8	7.1	2.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.4	0.3	2.7	1.1	3.8	6.4	4.6	14.5	15.0	26.2	14.3	9.3	1.3	0.1	0.0	0.0	0.0
13	0.0	0.0	0.0	0.3	2.0	1.4	2.3	2.9	1.7	10.3	9.2	22.9	24.4	19.8	2.6	0.3	0.0	0.0	0.0
14	0.0	0.0	0.2	0.0	0.1	0.2	0.4	0.3	2.0	3.2	3.9	13.0	18.3	37.9	18.1	2.4	0.2	0.0	0.0
15	0.0	0.0	0.0	0.0	0.4	0.0	0.4	1.2	1.2	3.5	2.7	7.0	12.8	29.8	24.8	11.6	2.7	0.0	1.9
16	0.0	0.0	0.0	0.0	0.9	0.0	0.0	0.0	0.0	1.8	0.9	2.6	6.1	21.9	43.0	14.0	7.0	0.9	0.9
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	2.0	10.2	18.4	14.3	42.9	0.0	8.2
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.7	7.7	7.7	0.0	7.7	0.0	23.1	0.0	38.5	0.0	7.7
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.7	0.0	6.7	13.3	0.0	26.7	0.0	46.7

Figure 3: Confusion matrix on the dev set

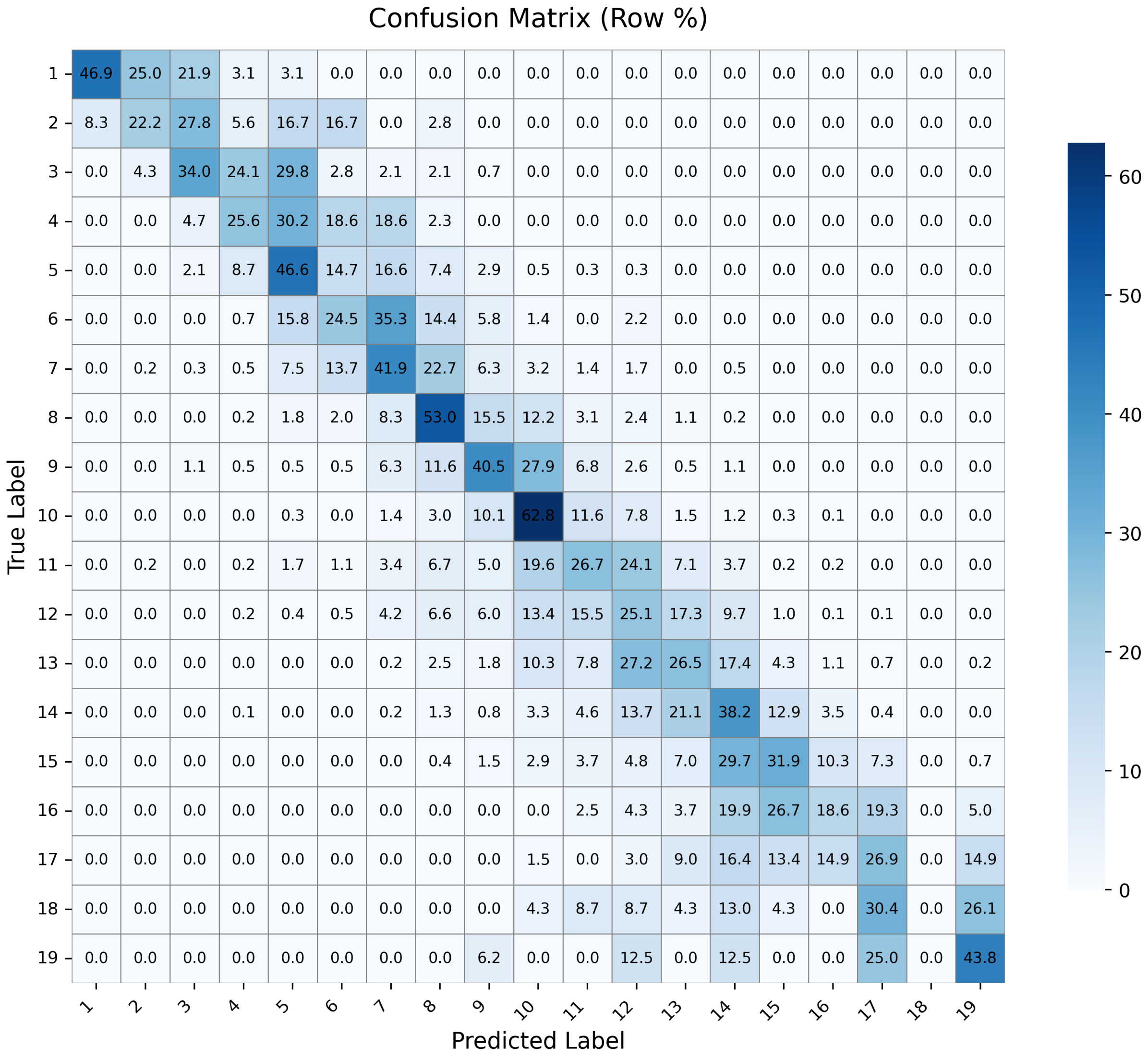


Figure 4: Confusion matrix on the public test set

D Model Hyperparamters

In this paper, we used the same hyperparameters for all models, training on the training set and tuning on the dev set. Each model was trained for 10 epochs with a batch size of 32 for both training and evaluation. We applied a weight decay of 0.01 and used a learning rate of 5e-5. The evaluation strategy was set to run at the end of each epoch, with the best model automatically loaded based on the lowest validation loss, which served as the metric for model selection.

Noor at BAREC Shared Task 2025: A Hybrid Transformer-Feature Architecture for Sentence-level Readability Assessment

Nour Rabih
Mohamed bin Zayed University of Artificial Intelligence
Nour.rabih@mbzuai.ac.ae

Abstract

This paper presents my participation in the Sentence-level Readability Assessment, Strict track of the BAREC Shared Task 2025 (Elmadani et al., 2025a). Building upon prior work that fine-tuned pre-trained transformer models (Elmadani et al., 2025b), this work explores the impact of incorporating a rich set of handcrafted features on readability prediction performance. A total of 51 features were extracted from the BAREC corpus (Elmadani et al., 2025b), including morphological, lexical, and syntactic indicators, leveraging established computational linguistics tools. These features were integrated into a hybrid architecture that combines transformer-based contextual embeddings with dense layers for feature processing. To optimize performance, experiments included freezing strategies and gradual unfreezing, alongside architectural variations with additional classification layers. Among the tested models, the best performance was achieved with MARBERT, reaching a Quadratic Weighted Kappa (QWK) of 80.95% on the test set, and 83.1% on the blind test set.

1 Introduction

Readability assessment aims to determine the ease or difficulty with which a reader can comprehend a given text. In educational contexts, accurate readability prediction supports tasks such as tailoring learning materials to students' proficiency levels, selecting appropriate reading passages, and developing adaptive learning systems. While research in English readability assessment has been extensive, Arabic remains comparatively underexplored, even though it has a rich morphology, diglossic nature, and complex orthographic and syntactic structures, all of which present unique challenges for computational modeling. The Sentence-level Readability Assessment task introduced in the BAREC Shared Task 2025 (Elmadani et al., 2025a) addresses these challenges by focusing on predicting 19 readability levels, based on the Taha/Arabi21 readability framework (Taha, 2017), for isolated Arabic sentences. Sentence-level assessment is inherently more challenging than document-level assessment, as the absence of broader discourse and contextual cues limits the available linguistic signals for prediction. Previous work (Elmadani et al., 2025b), has demonstrated competitive performance using fine-tuned transformer models without incorporating additional features. In this work, we present a hybrid approach that integrates 51 handcrafted linguistic and structural features with transformer-based contextual embeddings. These include counts of specific morphological forms (e.g., dual and plural noun/adjective inflections, broken plurals, verb tense and voice distinctions), syntactic constructions (e.g., nominal and verbal sentence types, complex clausal structures, object presence), functional particles (e.g., negation, prepositions, demonstratives, vocatives), and broader lexical indicators (e.g., unique word count, content word proportion, vocabulary richness measures. We further explore strategies such as layer freezing, gradual unfreezing, and the addition of extra classification layers to enhance performance. The results on both the test and blind test sets demonstrate that the inclusion of complementary features alongside transformer representations can yield improvements over purely transformer-based baselines, though the degree of improvement is model-dependent.

2 Background

The BAREC Shared Task 2025 (Elmadani et al., 2025a) introduced the Sentence-level Readability Assessment challenge for Arabic, designed to promote the development of models capable of fine-grained readability prediction. The task is framed as a multi-class classification problem with 19 dis-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 331–342
November 8-9, 2025 ©2025 Association for Computational Linguistics

crete readability levels. These levels are assigned to individual sentences based on a combination of linguistic, lexical, and pedagogical criteria, enabling precise targeting of reading materials to learner proficiency levels. I participated in the Strict Track where participants are restricted to using only the provided training data without incorporating any external corpora or embeddings. The dataset is split into training, development, test, and blind test sets. Each instance comprises an Arabic sentence and its readability label. The challenge lies in the granularity of the classification (19 levels), the diglossic and morphologically rich nature of Arabic, and the limited contextual cues available at the sentence level.

While readability assessment for English has benefited from decades of research using both handcrafted features and neural models (Sun et al., 2020; Deutsch et al., 2020; Heilman et al., 2007; Petersen and Ostendorf, 2009), Arabic-specific efforts remain comparatively limited. Early efforts relied on textbook corpora and statistical machine learning models (Al-Khalifa and Al-Ajlan, 2010). More recent work has explored both handcrafted linguistic features and modern pretrained language models (PLMs) such as AraBERT (Berrichi et al., 2024). The latest research trends emphasize hybrid approaches that combine traditional rule-based methods with PLMs, leveraging their complementary strengths for improved Arabic readability prediction (Liberato et al., 2024).

The SAMER project has further advanced Arabic readability resources. The SAMER Lexicon (Al Khalil et al., 2020) provides a five-level readability-annotated lexicon of approximately 26K lemmas, covering multiple dialects and achieving high inter-annotator agreement. Building on this, the SAMER Corpus (Alhafni et al., 2024) constitutes the first manually annotated Arabic parallel corpus for text simplification, consisting of around 159K words from 15 Arabic novels, each accompanied by two simplified parallel versions at different readability levels. These resources provide an important foundation for readability and simplification research in Arabic.

In addition, (Hazim et al., 2022) introduced a Google Docs add-on for Arabic word-level readability visualization. The tool integrates the SAMER Lexicon with morphological analysis and Arabic WordNet to highlight difficult words in context and suggest simpler alternatives. This practical interface enables annotators and educators to assess, simplify, and edit text directly within a familiar document editor, thereby making readability resources more accessible and actionable for corpus creation and pedagogical tasks.

3 System Overview

In this paper, the system adopts a hybrid architecture that integrates transformer-based contextual embeddings with handcrafted linguistic features for sentence-level readability prediction in Arabic. The design was motivated by the need to capture both deep semantic representations and explicit linguistic signals grounded in the BAREC annotation framework (Habash et al., 2025).

Five pre-trained models were experimented with: MARBERT , MARBERTv2 (Abdul-Mageed et al., 2021), AraBERTv2, AraBERTv02 (Antoun et al., 2020), and CamelBERT-MSA (Inoue et al., 2021). These models were selected for their strong performance on Arabic NLP tasks and their coverage of Modern Standard Arabic (MSA) and dialectal variants. For each model, the final hidden state of the [CLS] token, was extracted as the sentence representation.

3.1 Features

To complement the transformer embeddings, we engineered 51 handcrafted features inspired by the BAREC guidelines (Habash et al., 2025). These include:

- **Morphological features:** counts of prefixes, suffixes, verb tenses, plural forms, passive voice, etc.
- **Syntactic features:** dependency-based indicators such as presence of nominal sentences, verbal sentences with/without objects, vocatives, preposed predicates, and coordination structures.
- **Word/syllable counts:** normalized counts of unique words and syllables, leveraging diacritized forms for accuracy.
- **Vocabulary-based features:** sentence-level lexical difficulty scores derived from a lemma–POS vocabulary dictionary, augmented with the SAMER (Al Khalil et al., 2020) and dialect-sensitive markers.
- **Content-based features:** estimated idea/conceptual difficulty levels (ranging

from concrete to symbolic/abstract), obtained by fine-tuning a sentence-level AraBERT (Antoun et al., 2020) classifier.

Full details of the features and their extraction process are provided in the appendix A, but a summary of the methodology is given here. Feature extraction combined a range of resources, including CAMeL Tools (Obeid et al., 2020), regular expressions, external lexicons, and custom Python scripts. Morphological features were obtained using the CAMeL Tools morphological disambiguator, which decomposed tokens into base forms and affixes. This enabled systematic counting of prefixes, suffixes, and clitics at the sentence level, as well as identifying verb tense and voice distinctions such as active versus passive forms. Broken plurals and feminine plurals were similarly detected through combinations of morphological tags, following the rules outlined in Table 4 in the appendix. Syntactic features were extracted from dependency parses generated by CamelParser (Elshabrawy et al., 2023). Each sentence was transformed into a syntactic tree, from which binary indicators were derived to mark the presence of grammatical phenomena such as nominal versus verbal sentences, vocatives, and coordination structures. This rule-based approach ensured that subtle markers of syntactic complexity were systematically encoded, as summarized in Table 5 in the appendix.

Content-based features followed the BAREC framework, which defines eight levels of conceptual difficulty from concrete ideas to abstract or symbolic knowledge. A sentence-level AraBERT (Antoun et al., 2020) classifier was fine-tuned to predict these levels, which were then used as categorical features. Vocabulary-based features were derived through a multi-step pipeline aimed at quantifying lexical difficulty. First, a lemma–POS dictionary was constructed from the BAREC training set by tracking the distribution of each pair across all 19 readability levels. To account for noise and rare outliers, three thresholding strategies were evaluated (strict, relaxed-1%, and relaxed-2%), where the relaxed-1% variant provided the best balance between robustness and sensitivity. This dictionary allowed each sentence to be assigned a vocabulary score based on the most advanced lemma–POS pair it contained. To further strengthen coverage and align with curriculum-based readability scales, the dictionary was enriched with entries from the SAMER lexicon (Al Khalil et al., 2020), which provided additional structured mappings between words and difficulty levels. Together, these methods ensured that the handcrafted features captured complementary dimensions of linguistic complexity-morphological, syntactic, semantic, and lexical-beyond transformer embeddings.

3.2 Hybrid Architecture

The system combines transformer embeddings with feature representations through a dual-branch architecture.

Transformer branch. A BERT encoder produces contextual embeddings for the input sentence.

Feature branch. Handcrafted features $\mathbf{f} \in R^d$ (where $d = 51$) are processed through a Multi-Layer Perceptron (MLP) with batch normalization and ReLU activations.

Fusion. The feature representation is concatenated with the transformer [CLS] embedding.

Classification. A linear layer (softmax) maps the fused representation to 19 readability levels.

4 Experimental Setup

4.1 Dataset and Splits

We conduct experiments on the BAREC Shared Task 2025 dataset (Elmadani et al., 2025b) , which provides labeled sentences for sentence-level readability assessment. Following the official setup, we use the train, development (dev), and test splits provided. Additionally, we evaluate the blind test set, which contains hidden labels released only for the final submission phase.

4.2 Preprocessing

We integrated both raw text and handcrafted features into our pipeline. For each split, we merged the sentence text with 51 extracted linguistic features. One-hot encoding was done on categorical features such as content and vocabulary related features. Finally, labels were shifted to a 0–18 range for compatibility with PyTorch's classification layer.

4.3 Training Setup

We experiment with five transformer models: **MARBERT, MARBERTv2, CamelBERT-MSA,**

Model	Development Set							Test Set						
	Acc19	±1	Dist	QWK	Acc7	Acc5	Acc3	Acc19	±1	Dist	QWK	Acc7	Acc5	Acc3
CamelBERT-MSA	46.36	61.60	1.47	72.97	56.81	62.80	70.18	48.39	64.27	1.35	75.37	58.81	63.71	71.21
MARBERTv2	52.64	67.91	1.23	78.40	62.38	67.10	73.80	53.23	68.49	1.16	80.06	62.65	66.98	73.24
AraBERTv02	45.31	60.82	1.58	67.22	55.75	62.04	68.91	46.73	63.01	1.48	68.80	56.82	62.15	69.13
AraBERTv2	43.42	59.86	1.53	72.37	53.98	60.94	68.00	44.99	62.65	1.41	74.29	55.56	61.05	68.76
MARBERT	**54.69**	**69.28**	**1.19**	**79.38**	**63.93**	**68.44**	**75.08**	**54.45**	**69.75**	**1.11**	**80.95**	**63.93**	**67.99**	**74.11**

Table 1: Sentence-level readability results on BAREC (Dev/Test). Best per column in **bold**.

	Acc19	±1	Dist	QWK	Acc7	Acc5	Acc3
Blind Set (submitted system)	56.10	72.50	1.00	83.10	67.00	70.50	75.80

Table 2: Official hidden-set results of our submission. Acc19 = exact 19-class accuracy; ±1 = adjacent accuracy; Dist = mean absolute distance.

AraBERTv2, and AraBERTv02. We use the Hugging Face Transformers library for model initialization and PyTorch for training. Tokenization is performed with the respective model's pretrained tokenizer, truncating or padding sequences to a fixed maximum length. Models were trained using AdamW (lr=2e-5), batch size 16, for 6 epochs with Cross-Entropy loss, linear warmup/decay scheduling, and 0.3 dropout on an NVIDIA CUDA-enabled GPU. To improve generalization, we adopt a gradual unfreezing strategy: BERT embeddings are frozen at the start, with the last 4 layers unfrozen after epoch 1 and the full encoder unfrozen after epoch 2. Early stopping with patience 3 is applied based on validation QWK.

4.4 Evaluation Metrics

Readability assessment is treated as an ordinal classification task. We adopt the official metrics of the shared task:

- **Quadratic Weighted Kappa (QWK)** – primary metric, penalizing larger misclassifications more heavily.
- **Accuracy (Acc19/Acc7/Acc5/Acc3)** – classification accuracy at different granularities (collapsing 19 labels into 7, 5, or 3 bins), as show in table 3.
- **Adjacent Accuracy (±1 Acc19)** – off-by-1 tolerance measure.

Granularity	Group	BAREC Levels (1-19)
Acc3	1	1–11
	2	12–13
	3	14–19
Acc5	1	1–7
	2	8–11
	3	12–13
	4	14–15
	5	16–19
Acc7	1	1
	2	2–5
	3	6–8
	4	9–10
	5	11–13
	6	14–15
	7	16–19

Table 3: Coarse-grained groupings of the 19 BAREC readability levels used to compute Acc3, Acc5, and Acc7.

5 Results

Table 1 reports the performance of the hybrid architecture on all five pretrained models on the development and test splits of the BAREC Shared Task 2025. Overall, MARBERT achieved the strongest performance, reaching a QWK of 79.38% on the dev set and 80.95% on the test set. It also achieved the lowest average distance (1.11) and the highest exact accuracy (54.45%), confirming its robustness for fine-grained sentence-level readability assessment. MARBERTv2 followed closely, with a test QWK of 80.06%, suggesting that both MARBERT variants are particularly well-suited for the task.

We compared the hybrid models to text-only baselines for each pretrained model from (**?**) . The feature branch produced consistent improvements only with MARBERT (best QWK and lowest distance), whereas CamelBERT-MSA and AraBERT (v2/v02) showed very similar scores with and without features across Acc^{19}, ±1 Acc, Dist, and QWK. This indicates that the benefit of feature–text fusion is model-dependent rather than universal, and that strong PLM representations can already capture much of the signal for some encoders.

Blind set (official leaderboard). On the blind set used for the leaderboard, our submitted system achieved the results in Table 2. This placed us **9th on the Strict Path**.

6 Conclusion

This work presented a hybrid transformer–feature architecture for sentence-level Arabic readability assessment in the context of the BAREC 2025 Shared Task. By integrating 51 handcrafted linguistic, syntactic, morphological, and lexical features with contextual embeddings from pretrained Arabic language models, the system sought to capture complementary signals for fine-grained readability classification across 19 levels. Experimental results highlighted that MARBERT delivered the strongest performance, achieving a QWK of 80.95% on the test set and 83.1% on the hidden leaderboard, underscoring its robustness for handling sentence-level complexity in Arabic. The findings demonstrate that while transformer-based models alone provide strong baselines, combining them with structured linguistic indicators can further enhance performance, though the degree of improvement is model-dependent. This work contributes valuable insights into how feature engineering and representation learning can be jointly leveraged for readability modeling in morphologically rich and diglossic languages like Arabic. Future research may focus on incorporating dialectal diversity, enriching the dataset with larger and more varied corpora, and further engineering linguistic features to capture nuanced aspects of Arabic sentence complexity.

7 Limitations

This work is constrained by several limitations that restrict the scope and generalizability of its findings. First, the observed benefits of combining handcrafted linguistic features with transformer-based embeddings appear to be model-dependent. Improvements were most notable with MARBERT, while other pretrained models showed less consistent gains. This raises questions about the robustness and generalizability of the hybrid approach, suggesting the need for broader experimentation across architectures and domains. Second, the current setup focuses on sentence-level readability assessment, which inherently overlooks discourse-level context. Cohesion, coherence, and pragmatic cues that extend beyond individual sentences are often crucial for determining text difficulty, and their absence limits the granularity of prediction. Third, although the handcrafted features were carefully designed to reflect BAREC annotation guidelines, they rely on rule-based extraction pipelines that may introduce errors or fail to capture more nuanced aspects of Arabic syntax, morphology, and dialectal variation. These constraints highlight the need for richer and more diverse datasets and the development of adaptive, data-driven feature engineering techniques. Addressing these challenges will be essential for advancing the accuracy, and real-world applicability of Arabic readability assessment systems.

Ethics Statement

While the author is affiliated with institutions linked to the shared-task organizers, no organizer was involved in the design, development, or evaluation of the systems presented in this paper. The work was conducted exclusively using the resources and information publicly released as part of the shared task, without any privileged access or special guidance. This statement is provided to clarify that no conflict of interest has influenced the reported results.

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 S Al-Khalifa and Amani A Al-Ajlan. 2010. Automatic readability measurements of the arabic text: An exploratory study. *Arabian Journal for Science and Engineering*, 35(2 C):103–124.

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the*

Twelfth Language Resources and Evaluation Conference, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

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.

Safae Berrichi, Naoual Nassiri, Azzeddine Mazroui, and Abdelhak Lakhouaja. 2024. Exploring the impact of deep learning techniques on evaluating arabic l1 readability. In *Artificial Intelligence, Data Science and Applications*, pages 1–7, Cham. Springer Nature Switzerland.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. *arXiv preprint arXiv:2006.00377*.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Ahmed Elshabrawy, Muhammed AbuOdeh, Go Inoue, and Nizar Habash. 2023. CamelParser2.0: A State-of-the-Art Dependency Parser for Arabic. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Nizar Habash, Hanada Taha-Thomure, Khalid Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Michael Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. 2007. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Human language technologies 2007: the conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 460–467.

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.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. 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 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Sarah E Petersen and Mari Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer speech & language*, 23(1):89–106.

Yuxuan Sun, Keying Chen, Lin Sun, and Chenlu Hu. 2020. Attention-based deep learning model for text readability evaluation. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

H. Taha. 2017. معايير هنادا طه لتصنيف مستويات النصوص العربية. دار الكتاب التربوي للنشر والتوزيع.

A Appendix

In this work, the Balanced Arabic Readability Evaluation Corpus (BAREC) is used as the primary dataset. The BAREC Annotation Guidelines (Habash et al., 2025) offer a detailed framework for readability annotation, considering six major linguistic dimensions: Spelling/Pronunciation, morphology, syntax, vocabulary, Idea/content, and word count. To align with this framework, a comprehensive set of linguistic features was engineered, rooted in these dimensions. These features fall under five main categories: morphological, syntactic, word/syllable counts, vocabulary-based, and content-based. Feature selection was guided by the criteria outlined in the BAREC guidelines to reflect the linguistic signals that influence sentence complexity.

Feature extraction was conducted using a combination of CAMeL Tools (Obeid et al., 2020), regular expressions, external lexicons, and custom Python scripts. Below is a breakdown of each feature group and the extraction methods used:

- **Morphological Features**

  - **Number of prefixes, suffixes, and clitics**: Extracted using CAMeL Tools' morphological disambiguator. Each token was decomposed into its base form and affixes, and counts were aggregated per sentence.
  - **Verb tense and voice (e.g., passive, active)**: Identified using the POS tags and morphological features provided by CAMeL Tools.
  - **Use of different forms (broken plurals, feminine plurals)**: Detected using morphological patterns and specific tag combinations (e.g.,singular form and plural num) from CAMeL analysis.

  Table 4 shows the specific rules for all morphological features.

- **Syntactic Features**
  Syntactic complexity plays a key role in determining sentence-level readability in Arabic. To capture this, a set of rule-based syntactic features was developed using dependency parsing outputs using the CamelParser(Elshabrawy et al., 2023).

  A dependency parse was first used to construct syntactic trees for each sentence, allowing for the identification of grammatical relations between words. From these structures, a set of binary features was extracted to reflect the presence or absence of key syntactic phenomena.

  Table 5 shows the specific rules for all Syntactic features.

- **Content-Based Features**
  The BAREC annotation guidelines include a dedicated dimension for evaluating the conceptual and semantic difficulty of a sentence, referred to as the content level. This dimension considers the type of knowledge required for comprehension, the presence of abstract or symbolic ideas, and the cognitive demands placed on the reader. The guidelines define eight content levels, ranging from direct and concrete ideas (e.g., daily life topics requiring no prior knowledge) to highly abstract, symbolic, or culturally nuanced content that assumes specialized background knowledge. To automatically estimate this content complexity, a sentence-level classifier was developed by fine-tuning an AraBERT model. The model was trained to predict one of the eight content levels defined in the guidelines, treating this as a multi-class classification task. These predicted levels were then included as features in the broader feature set used for readability prediction. Table 6 provides a summary of the eight content levels defined in the BAREC framework, along with example indicators used during annotation.

- **word/syllable counts**

  - **Word count**: Computed as the number of unique words in a sentence, ignoring repetitions, or punctuation.
  - **syllable count**: The number of syllables in each word is computed by incorporating morphological and phonetic information. The CAPHI (consonant–vowel pattern) representation, the diacritized form of the word, and morphological prefix annotations are used for a more accurate count of syllables. The CAPHI string is tokenized and scanned for vowel segments, each indicating a potential syllable. Specific linguistic rules are applied to refine the syllable count:
    - The final vowels are excluded if it is a diacritic (حركات الإعراب).
    - Morphological prefixes such as the definite article (ال التعريف) and conjunction (واو عاطفة) are excluded, as they do not contribute to the core syllabic structure of the main word.

ccc

Feature	Feature (Arabic)	Rule
Singular imperfective verb	الفعل المضارع المفرد	num=s, asp=i, pos=verb
Prtoclitic: Definite article Al+	سوابق: ال التعريف	prc0=Al_det
Proclitic: Conjunction wa+	سوابق: واو العطف	prc2=wa_conj
Enclitic: First Person Singular pronoun	لواحق: ضمير المتكلم المفرد المتصل	enc0=1s_pron / 1s_poss / 1s_dobj
Plural imperfective verb	الفعل المضارع الجمع	pos=verb, asp=imp, num=p
Prepositional proclitics	سوابق: حروف جر متصلة	prc1=bi_prep / li_prep / ka_prep
Enclitic: Singular and Plural pronouns	لواحق: ضمير متصل مفرد أو جمع	enc0 in [1p_dobj, ..., 3p_pron]
Dual (in nouns and adjectives)	المثنى في الأسماء والصفات	num=d, pos=noun / adj / noun_quant / adj_comp
Sound feminine plural	جمع المؤنث السالم	form_num=p, form_gen=f, pos=noun / adj
Singular and plural perfective verb	الفعل الماضي المفرد والجمع	pos=verb, asp=p, num=s / p
Sound masculine plural	جمع المذكر السالم	form_gen=m, form_num=p, pos=noun / adj
Dual perfective verb	الفعل الماضي المثنى	asp=p, num=d, pos=verb
Dual imperfective verb	الفعل المضارع المثنى	asp=i, num=d, pos=verb
Singular imperative verb	فعل الأمر المفرد	pos=verb, asp=c, num=s
Enclitics: dual pronoun	لواحق: ضمير المثنى المتصل	enc0=[2d_dobj, ..., 3d_pron]
Broken plurals	جمع التكسير	pos=noun / adj, form_num=s, num=p
Waw of oath	واو القسم	prc2=wa_prep and followed by qassam_lex
Plural imperative verb	فعل الأمر الجمع	asp=c, num=p, pos=v
Conjunctions (e.g., then, until, or...)	أدوات ربط	match of lex
Dual imperative verb	فعل الأمر للمثنى	asp=c, num=d, pos=verb
Ba of oath	باء القسم	prc1=bi_prep, lex in qassam_lex
Passive voice	المبني للمجهول	vox=p, pos=verb
Ta of oath	تاء القسم	prc1=ta_prep, lex in qassam_lex

Table 4: Morphological Features and Rules from BAREC Guidelines

Table 5: Syntactic Features and Rules from BAREC Guidelines

Feature	Feature (Arabic)	Rule
Nominal sentence	الجملة الاسمية	parent != inna and sisters, has a child with Dependency relation: SBJ (subject), POS tag not equal to VRB
Verbal sentence w/o direct object	جملة فعلية بدون مفعول به	parent = verb, no OBJ Dependency relation
Preposition and object	جار+مجرور	parent pos: PRT, pos=prep in FEATS, has a child with Dependency relation: OBJ
Verbal sentence with one nominal direct object	جملة فعلية مع مفعول به واحد اسم	parent = PRT with pos=prep, has a child with Dependency relation: OBJ
Sentence with two verbs	جملة فيها فعلين	verb count
Verbal sentence with a clausal direct object introduced with Masdar 'an [˜to/that]	جملة فعلية مفعولها أن المصدرية	Token is , pos = PRT Has a child: pos = VRB deprel = OBJ asp=i(imperfective)
Verbal sentence with two direct objects	جملة فعلية تتعدي إلى مفعولين	parent pos = VRB Two children with Dependency relation = OBJ
Vocative	المنادى	parent has pos:PRT, FEATS include pos=part_voc Has a child with Dependency relation = OBJ
Inna and its sisters	إن وأخواتها	parent matches the lemma set, has a child with Dependency relation=PRD
Kana and its sisters	كان وأخواته	parent pos=verb, lemma in kana set, has a child with Dependency relation = PRD
Preposed predicate, postponed subject	الخبر المقدم والمبتدأ المؤخر	Dependency relation= SBJ, has a child: pos != VRB, index of parent < index of child
Nominal sentence with a nominal predicate	جملة أسمية خبرها جملة أسمية (فيها مبتدآن)	The sentence does not start with a verb It contains a child node with deprel == "TPC" (topic)
False idafa (tall in stature)	إضافة خيالية (لفظية)	parent pos: NOM, pos=adj in FEATS, Has a child with Dependency relation = IDF
Exception	استثناء	pos '='part$_{r}estrict$

Table 6: Idea / Content Levels in English and Arabic

Idea / Content	فكرة ومحتوى
Direct, explicit, and concrete idea. No symbolism in the text.	فكرة مباشرة وصريحة وحسية. لا رمزية في النص.
Content is from the reader's life. No symbolism in the text.	المحتوى من حياة القارئ. لا رمزية في النص.
Some symbolism, or not everything is stated directly in the sentence.	بعض الرمزية أو عدم التصريح المباشر بكل المقصود في الجملة
Some symbolism that requires the reader to seek help to understand the idea.	بعض الرمزية يحتاج معها القارئ إلى مساعدة من يشرح له المقصود من الفكرة
Some symbolism at the event level in the sentence that the reader understands through prior knowledge.	هناك شيء من الرمزية على مستوى الحدث في الجملة يدركها القارئ بنفسه أو من خلال معارفه السابقة
A degree of symbolism and a need for prior knowledge to understand the meaning of the sentence.	هناك درجة من الرمزية وحاجة للمعرفة السابقة كي يُفهم المقصود من الجملة.
Symbolic ideas and deeper meanings, especially in terms of the psychological dimension of characters/events. Local cultural expressions that may not be understood by those outside the culture.	أفكار رمزية ومعنى باطن خاصة على صعيد البعد النفسي للشخصيات أو الأحداث. تعابير ثقافية محلية قد لا يفهمها من لا يشترك في نفس الثقافة.
Symbolic, abstract, scientific, or poetic ideas that require prior linguistic and cognitive knowledge to understand.	أفكار رمزية، مجردة، عِلمية، أو شعرية و تحتاج إلى معارف لغوية ومعرفية سابقة للبناء عليها لأجل فهمها.

        ∗ In the absence of CAPHI information, syllables are counted by identifying diacritic characters corresponding to short vowels within the diacritized form of the word.

- **Vocabulary-based Features**

  Vocabulary was handled in three different ways to estimate the lexical difficulty of sentences. Firstly, to estimate the vocabulary difficulty of a sentence, a level-based vocabulary scoring system was constructed using the training set of the BAREC dataset. The process began by extracting all lemma–Part of Speech (POS) pairs from the training data using the cameltools disambiguator. For each pair, the number of occurrences was counted across all 19 BAREC readability levels. This allowed for identifying the earliest level at which each lemma–POS pair appeared in the corpus.

  To account for annotation noise or occasional use of advanced vocabulary in lower levels, three variants of vocabulary level assignment were considered:

  - **Strict**: The lowest level at which the lemma–POS pair appeared.
  - **Relaxed (1%)**: The lowest level where the pair appeared, allowing for a 1% error margin of frequency across levels.
  - **Relaxed (2%)**: Similar to the above but with a 2% margin.

  These thresholds introduced flexibility, ensuring that a few early occurrences of complex vocabulary in lower-level sentences did not skew the overall difficulty estimation.

  Once each lemma–POS pair was associated with a level, a vocabulary-level dictionary was constructed containing all lemma–POS pairs from the training data along with their assigned difficulty levels. This dictionary was then used to map vocabulary in the development and test sets. New input sentences were transformed into lists of lemma–POS pairs, and for each sentence, the vocabulary level was defined as the highest (i.e., most difficult) level among all matched pairs. Experiments were conducted using all three threshold variants, and the version yielding - 1% error margin- the best performance was selected for use in the final model.

  In addition to the data-driven vocabulary extracted from the training set, it was observed that expanding the lexical coverage further improved performance. The BAREC annotation guidelines specifically reference certain levels from the SAMER readability lexicon (Al Khalil et al., 2020) as indicative of vocabulary difficulty. To incorporate this, the SAMER lexicon was used to augment the existing vocabulary-level dictionary. Lemma–POS pairs from SAMER were assigned levels in accordance with the BAREC guidelines, thereby enriching the vocabulary feature set with structured, curriculum-aligned information.

  To introduce dialectal sensitivity—also highlighted in the BAREC guidelines—a supplementary lexicon from the BAREC project was utilized. This lexicon consists of approximately 5,000 annotated words, each marked with a dialectal match indicator. Although this represents a relatively small subset of the overall vocabulary, it introduces an important dimension of variation and adds a foundational layer of dialectal awareness to the feature set.

  This layer is particularly valuable because it enables the model to distinguish between vocabulary that overlaps across Modern Standard Arabic and dialects versus vocabulary that exists only in dialectal usage. Words that are common across both MSA and dialects—such as “chair" (كرسي), which appears consistently in both—are typically introduced at earlier reading levels and thus ranked lower in complexity. In contrast, words like “window," which differ in MSA and dialectal forms (e.g., “نافذة" vs. “شباك"), are treated as more complex and are ranked at higher readability levels. Incorporating this information allows the model to better reflect the lexical difficulty that dialectal divergence introduces, especially for learners who are trained primarily on MSA vocabulary.

  In addition to the above features, the barec dataset specifes certain closed groups of vocabs that can be identified using the Cameltools disambiguator, these are shown in table 7.

Table 7: Vocabulary Feature Levels (English and Arabic)

Vocabulary	المفردات
Proper noun Personal pronouns (non-clitics)	اسم علم ضمير منفصل
Singular demonstrative pronoun	اسم الإشارة المفرد
Prepositions	حروف الجر
Dual and plural demonstrative pronoun	اسم اشارة مثنى، جمع
Negation particles	أحرف النفي
Singular relative pronouns	أسماء الوصل المفردة
Dual and plural relative pronouns.	أسماء الوصل المثنى والجمع

PalNLP at BAREC Shared Task 2025: Predicting Arabic Readability Using Ordinal Regression and K-Fold Ensemble Learning

Mutaz Ayesh
Cardiff University / Cardiff, Wales, UK
AyeshMA@cardiff.ac.uk

Abstract

PalNLP addressed Arabic readability level prediction as a fine-grained ordinal classification problem by strictly using the Balanced Arabic Readability Evaluation Corpus (BAREC). The approach treats the 19-class ordinal classification problem as a regression task with post-hoc threshold optimization, leveraging a BERT-based model and an ensemble strategy. The system achieved a Quadratic Weighted Kappa (QWK) score of 81.1 in the blind test dataset, indicating an almost perfect agreement between the system's classifications and the true labels, and placing 18th out of 24 teams. The findings show that the model effectively learned broad readability patterns, with a competitive ±1 accuracy, but faced challenges in accurately predicting readability levels of most sentences.

1 Introduction

The overlap between automatic readability assessment (ARA) and other NLP tasks highlights its importance. In summarization, for example, readability frameworks and ARA may complement classic summarization metrics by evaluating the output of audience-aware or level-controlled summarization models by predicting the level of the generated summary against the original text input. Controlling summaries for readability levels can help these models generate summaries that are more suitable for their targets, as was done by Luo et al. (2022) for biomedical texts.

Similarly, Plain Language (PL) and Easy-to-Read[1] (E2R) initiatives have been gaining traction in Europe (Espinosa-Zaragoza et al., 2023; Martínez et al., 2024; Madina et al., 2024). They aim to make governmental texts more accessible for non-native speakers, people with reading limitations, and people with cognitive, intellectual, or learning disabilities. As part of the CLEARS Shared Task in IberLEF-2025 (Botella-Gil et al., 2025), Ayesh et al. (2025) attempted to transform Spanish texts in accordance with PL and E2R guidelines and used the Fernández Huerta Readability Index as one of the main metrics of evaluating the results. This index shows the importance of readability levels as an evaluation metric for a successful summary. Such alignment to reader proficiency supports better comprehension and learning outcomes (Elmadani et al., 2025b).

This task is particularly challenging for Arabic due to its morphological richness and orthographic ambiguity, and the *diglossia* that exists between Modern Standard Arabic and spoken dialects (Suwaiyan, 2018; Liberato et al., 2024; Elmadani et al., 2025b). The scarcity of large, fine-grained, and publicly available Arabic readability resources has further limited the development of robust modeling approaches. Existing resources like the word-level SAMER Lexicon (Al Khalil et al., 2020) and word- and document-level SAMER Corpus (Alhafni et al., 2024) are valuable but often domain-specific or coarse in granularity.

The new Balanced Arabic Readability Evaluation Corpus (BAREC) (Elmadani et al., 2025b) offers an opportunity to explore readability prediction with high granularity by providing over 69 thousand sentences[2] labeled across 19 readability levels, enabling modeling that captures lexical, morphological, and syntactic variation.

This paper presents the system that was submitted to the BAREC 2025 Shared Task (Elmadani et al., 2025a) for predicting BAREC readability levels, which can be summarized as a regression-then-discretization approach that is optimized for Quadratic Weighted Kappa (QWK). This formulation directly accounts for the ordinal nature of the labels and prioritizes proximity to the real level over exact level matches. The contributions of

[1] Easy-to-read is also referred to as "easy reading".

[2] *Sentence* here is used broadly to mean a standalone text.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 343–349
November 8-9, 2025 ©2025 Association for Computational Linguistics

this system can be summarized as follows: (1) a regression-based approach with coordinate descent threshold optimization for ordinal classification, (2) the integration of a BERT-based model with class imbalance handling and ensemble aggregation, and (3) the analysis of performance across granularities, showing strong ordinal capture but challenges in fine-grained separation.

2 Background

Readability assessment in Arabic has benefited from recent advances in corpus creation and lexical resource development. The Taha/Arabi21 framework (Taha-Thomure, 2017) provides a 19-level scale for educational text leveling, which BAREC adapts to the sentence level through refined annotation guidelines encompassing lexical, morphological, syntactic, and semantic features (Habash et al., 2025). Complementary resources include the SAMER readability lexicon (Al Khalil et al., 2020), which contains over 26,000 lemmas annotated with five difficulty levels by language experts from multiple Arab regions, and the SAMER reading corpus (Alhafni et al., 2024), which spans 1.4 million tokens from UAE curriculum materials and 5.6 million tokens from literary works. These resources support both lexical- and document-level readability modeling. Tools such as the word-level readability visualization add-on (Hazim et al., 2022) demonstrate practical applications in assisted text simplification and highlight the potential of integrating lexical difficulty features into automatic assessment systems.

In this shared task, participants predicted readability levels of texts from the BAREC dataset, with evaluation based on QWK. PalNLP participated in the *strict*, *sentence-level* track, meaning no additional external data was used in the development of the system alongside the sentence-level version of the BAREC dataset.

3 System Overview

The system addresses Arabic readability prediction as a continuous regression problem with post-hoc threshold optimization, treating the 19-class ordinal classification task through a regression-then-discretization approach optimized for Quadratic Weighted Kappa (QWK). This is due to the ordinal nature of the readability levels. The system used CAMeL-Lab's readability-arabertv2-d3tok-CE, which was used in the dataset's paper (Elmadani et al., 2025b), as the foundation model. Although the model was originally fine-tuned as a classification model with cross-entropy loss, this system adapted its architecture for regression to leverage the strong readability-sensitive features learned in the CE setup while optimizing for continuous predictions. It was then combined with a threshold optimization algorithm, and later, an ensemble methodology.

The system used the sentence-level BAREC dataset, loaded from HuggingFace, without any additional data. Instead of using the default training and validation splits, these two sets were combined, and 5-fold stratified cross-validation was applied to the merged dataset. This was due to a sustained plateau in validation loss throughout the initial experiments. As a result, the system is not directly comparable to other participants' systems. The test split remained unchanged. To address class imbalance, PyTorch's WeightedRandomSampler was used with inverse class frequency weighting during training to ensure that rare readability levels were adequately represented.

Hyperparameter	Value
Input processing	Padding to 512 tokens
Batch size	16
Epochs	6 with early stopping
Learning rate	2e-5 with adaptive scheduling

Table 1: Hyperparameters used in the system. Early stopping also includes patience of 3 epochs.

4 Experimental Setup

The core architecture consists of a BERT-based regressor with a single continuous output head where ordinal class labels are treated as continuous values for training. MSE loss was employed with AdamW optimization, and a combination of linear warm-up and ReduceLROnPlateau scheduling based on validation QWK performance. Table 1 shows the specific hyperparameter values in the system.

Throughout the tens of experiments that were run before this final one was adopted, the systems under-performance on the validation dataset was observed despite achieving good scores in the training. A key innovation in this approach is the coordinate descent algorithm for threshold optimization; rather than using simple rounding to discretize continuous predictions, the model iteratively optimizes the thresholds associated with each class to maximize QWK on validation data through grid-based coordinate descent with multiple passes. This strategy consistently provided 1-2% improvements over naive rounding during the training.

Final results are derived by thresholding the continuous predictions into class labels. The best model of each fold gets saved and the different folds are used to predict the readability level by aggregating the predictions using each fold's threshold weights.

5 Results and Error Analysis

5.1 On the provided datasets

The results of cross-validation, found in Table 4 in Appendix A, showed consistency, with a QWK range of 79.85-80.21, indicating robust generalization.

After the training was done, and the system concluded with a QWK score of 79.66 with global thresholds, the predictions on the provided test dataset were obtained by ensembling all different folds, where predictions from the best model of each fold were combined using a weighted ensemble approach. This means that fold-specific threshold weights were applied before aggregating to final discrete readability predictions. The final QWK score on the test set was 77.7.

Table 2 summarizes the system's performance on the test dataset after ensembling. The results show that the model certainly learned the ordinal structure of BAREC well and that its misclassified labels were close to the correct level, as evidenced by the ±1 level accuracy. The model, however, struggled with exact classification. An illustration of this can be found in Figure 1.

Impact of domain and word count. After a curious look into the top 100 sentences with the predicted levels furthest from the true levels[3], it was apparent that those that were underestimated (i.e., the true readability levels were higher than the predicted ones) were short, with 94% of those being fewer than 5 words long. 76% of those short sentences are specialized or advanced texts; 32% are specialized and advanced texts from the Emirati curriculum, while 22% come from the Quran. Detecting the true readability level of these specific sentences might have required a model that also considers qualitative features, such as the source of the text and its class. Examples of such texts can be found in Appendix D.

A similar pattern can be seen among sentences whose readability levels were overestimated (i.e., their true readability levels were lower than the predicted ones) where 46% were 5 words long or fewer, and 68% were 7 words long or fewer. The length of these sentences might have had an impact, but the impact of the type of the text (foundational, specialized, or advanced) was not as significant, as there was somewhat an equal distribution between specialized and advanced (52%) and foundational (48%) texts. A deeper look into why the model overestimated their levels is required.

Impact of diacritics. Despite using an Arabic-specific BERT model, it seems that the system continuously misclassified texts with diacritics as ones with high readability levels. While the reasons behind why that happened make sense, it was not an outcome that was expected at all. The sentence with one of the greatest differences from the true readability level was a diacritized proper name[4] with no inherent difficulty. It had a readability level of 3 but was misclassified as having a readability level of 15. Another example[5] had a readability level of 8 but was classified as 15 due to the diacritics. These stark differences reflect the importance of pre-processing Arabic texts to allow the trained models to capture real features that reflect the readability levels of texts, rather than superficial ones such as diacritics that do not necessarily entail a difficult or advanced level.

After this error was detected, the test set was passed through the system to generate predictions, however, this time the diacritics were stripped using PyArabic's[6] `strip_diacritics` method beforehand. The performance on the de-diacritized test set can be found in Table 2, alongside the original scores before stripping diacritics. The new results better resemble those of PalNLP's on the blind test set, and an improvement can be seen in all metrics, especially a +3.5 improvement in the QWK score and both the exact and ±1 level accuracy scores.

Additionally, the ranges of difference between the true readability and predicted levels dropped from (-15, 12) to (-11, 8)[7]: the drop in each com-

[3] 50 sentences in each direction (positive and negative differences) were considered in this analysis.

[4] شِهابُ الدِّين أَحْمَدُ بْنُ ماجِدٍ
Sentence ID: 10400320088

[5] عَجَبًا مِن النظَّارةِ السوداءِ لم تَحجُبِ المعنى عن الرُّقباءِ
Sentence ID: 30100250057

[6] https://pypi.org/project/PyArabic/

[7] The highest negative difference is on the left, and the highest positive difference is on the right.

Metric	Before SD	After SD
QWK	77.7	81.25
Exact Accuracy	29.99%	34.37%
±1 Level Accuracy	65.88%	69.48%
7-Class Accuracy	50.96%	54.78%
5-Class Accuracy	51.85%	53.17%
3-Class Accuracy	66.58%	67.94%

Table 2: The system's performance on the test set, before and after stripping diacritics (SD).

ponent indicates reduced error bounds, reflecting fewer extreme under- and over-estimations and more tightly aligned predictions with the true readability levels. Table 5 in Appendix B further solidifies the improvement in performance; it shows a great improvement in exact predictions (+319) coupled with consistently less differences after stripping diacritics.

The heat maps in Appendix D further illustrate the improvement: after stripping diacritics, the confusion matrix becomes more diagonal, with noticeably fewer misclassifications concentrated in the upper readability levels.

5.2 On the blind test dataset

The system achieved 18th place out of 24 teams with an official QWK score of 81.1. The score is close to the organizers' baseline of 81.5. Table 3 contains a summary of the performance of PalNLP's system on the blind test set. Overall, the consistency between cross-validation (79.96-80.21), test set (77.7), and competition results (81.1) demonstrates the effectiveness of the validation strategy, and the system performing better in the blind test set shows that the model did not overfit on the training dataset.

It can be safely said that the ordinal structure of the BAREC dataset was effectively captured by the system, as evidenced by the much smaller gap in ±1 accuracy between the system (69.8%) and the organizers' (72.0%). This indicates that the model learned the ordinal structure well and that its misclassified labels are mostly close to the correct level. Additionally, performance gaps between PalNLP's system and the baseline decreased dramatically as classification granularity was reduced, from 25% difference in 19-class accuracy to only 4% in 3-class accuracy. This shows that the model successfully learned broad readability patterns.

The system, however, struggled with fine-grained distinctions between adjacent levels. The significant gap in exact accuracy between this system (33.1%) and the organizers' (58.1%) contrasted with the minimal QWK difference is expected as the regression framework was optimized for rank correlation rather than precise classification.

Metric	PalNLP	Baseline
Avg. Absolute Distance	1.3	1.0
QWK	81.1	81.5
Exact Accuracy	33.1%	58.1%
±1 Level Accuracy	69.8%	72.0%
7-Class Accuracy	57.2%	67.7%
5-Class Accuracy	63.6%	71.4%
3-Class Accuracy	72.5%	76.5%

Table 3: The system's performance on the blind test set, provided by the prediction log on CodaBench. The baseline scores were taken from the competition's leaderboard on CodaBench.

6 Conclusion

This paper presented a regression-then-discretization system for Arabic readability prediction on the BAREC dataset, with a focus on maximizing QWK. By modeling the task as a continuous regression problem with post-hoc threshold optimization, the results showed that the system captured the ordinal nature of readability levels in BAREC, favoring proximity to the true label over exact agreement. The BERT-based model with stratified cross-validation, class imbalance handling, and ensemble aggregation produced results that consistently generalized across validation, test, and competition evaluations.

Several key observations emerged. (1) The system broadly understood readability patterns but found fine-grained separation between adjacent levels challenging. (2) The regression formulation, combined with threshold optimization, consistently outperformed naive rounding strategies and improved alignment with the dataset's ordinal structure. (3) The error analysis highlighted systematic weaknesses, such as underestimation of short, specialized sentences, and misclassification of diacritized text as advanced-level material. (4) The consistent alignment between cross-validation, test, and blind test results prove that PalNLP's strategy was robust with minimal overfitting.

The role of pre-processing and text-specific features point toward future refinements, such as training the model *after* handling diacritics and possibly other pre-processing techniques. Further work may explore the performance of this system after integrating additional resources such SAMER that can alleviate the effect of class imbalances in BAREC.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Mutaz Ayesh, Nicolás Gutiérrez-Rolón, and Fernando Alva-Manchego. 2025. CardiffNLP at CLEARS-2025: Prompting large language models for plain language and easy-to-read text rewriting.

Beatriz Botella-Gil, Isabel Espinosa-Zaragoza, Alba Bonet-Jover, Margot Madina, Lucas Molino Piñar, Paloma Moreda, Itziar Gonzalez-Dios, María Teresa Martín Valdivia, and Ureña. 2025. Overview of CLEARS at IberLEF 2025: Challenge for Plain Language and Easy-to-Read Adaptation for Spanish Texts. *Procesamiento del Lenguaje Natural*, 75.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A Large and Balanced Corpus for Fine-grained Arabic Readability Assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Isabel Espinosa-Zaragoza, José Abreu-Salas, Paloma Moreda, and Manuel Palomar. 2023. Automatic text simplification for people with cognitive disabilities: Resource creation within the ClearText project. In *Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability*, pages 68–77, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2022. Readability controllable biomedical document summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4667–4680, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Margot Madina, Itziar Gonzalez-Dios, and Melanie Siegel. 2024. Towards reliable e2r texts: A proposal for standardized evaluation practices. In *Computers Helping People with Special Needs*, pages 224–231, Cham. Springer Nature Switzerland.

Paloma Martínez, Lourdes Moreno, and Alberto Ramos. 2024. Exploring large language models to generate easy to read content. *Preprint*, arXiv:2407.20046.

Laila Suwaiyan. 2018. Diglossia in the arabic language. *International Journal of Language Linguistics*, 5.

Hanada Taha-Thomure. 2017. *Arabic Language Text Leveling* معايير هنادا طه لتصنيف مستويات النصوص العربية. Educational Book House دار الكتاب التربوي للنشر والتوزيع.

A Cross-validation scores

Table 4 presents the five-fold cross-validation results. Across folds, the optimized thresholding strategy (QWK$_{opt}$) consistently outperformed fixed rounding (QWK$round$) by about 1–2 points, confirming the benefit of post-hoc threshold optimization. Training generally converged within 3–6 epochs, with early stopping triggered in three out of five folds. These results indicate stable model performance and reduced overfitting across folds.

B Differences between predicted and true levels in the test set, before and after SD

Table 5 shows the distribution of differences between predicted and true levels before and after stripping diacritics. The results show a reduction in large deviations (e.g., no cases at ±15 or ±12 after SD, and consistent decreases from ±11 to ±5), alongside an increase in exact matches (0 difference rose from 2185 to 2504). This indicates that SD reduces the frequency of extreme cases while improving overall alignment with the gold labels.

C Heat maps

Figures 1 and 2 show the normalized confusion matrices before and after stripping diacritics. The post-SD heat map exhibits a clearer diagonal pattern, reflecting reduced over-prediction of lower readability levels and stronger agreement between true and predicted labels.

Fold	**QWK**$_{opt}$	**QWK**$_{round}$	**Epochs**	**ES**
1	79.85	78.05	3	At epoch 1
2	80.21	78.82	5	At epoch 3
3	79.96	78.56	6	No
4	79.96	78.85	6	No
5	79.90	78.77	6	At epoch 5

Table 4: Cross-validation results, with 5 folds. QWK$_{opt}$ refers to the QWK score using the threshold optimization strategy detailed earlier, as opposed to the score using fixed rounding shown in QWK$_{round}$. Early stopping (ES) was included here to show when the QWK results on the (custom) validation dataset plateaued.

Difference	**F**$_{beforeSD}$	**F**$_{afterSD}$
±15	1	0
±12	1	0
±11	5	2
±10	6	3
±9	11	6
±8	27	18
±7	84	62
±6	94	59
±5	185	163
±4	344	306
±3	600	519
±2	1128	1086
±1	2615	2558
0	2185	2504

Table 5: The frequencies of differences between the predicted and true levels in the test set, before and after stripping diacritics (SD). The 0 difference in the last row is synonymous with the frequency of exact predictions made by the system.

D Examples of extreme differences

- نُمُوُّ السُّكّانِ "Population growth"
  predicted RL: 6, true RL: 14
  (ID: 20400120059)

- إِضاءَةٌ "Lighting"
  predicted RL: 3, true RL: 11
  (ID: 20400200031)

- .القانونُ "The law."
  predicted RL: 4, true RL: 12
  (ID: 20400360004),

- :أُناقشُ "I discuss"
  predicted RL: 4, true RL: 12
  (ID: 20400550017)

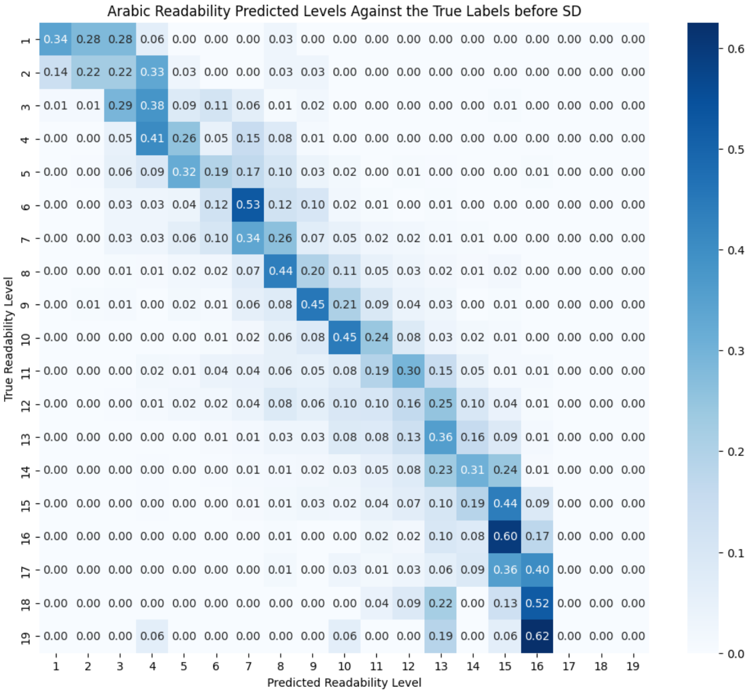


Figure 1: A normalized confusion matrix (heat map) of predicted levels against the true levels of texts in the test dataset **before** stripping diacritics.

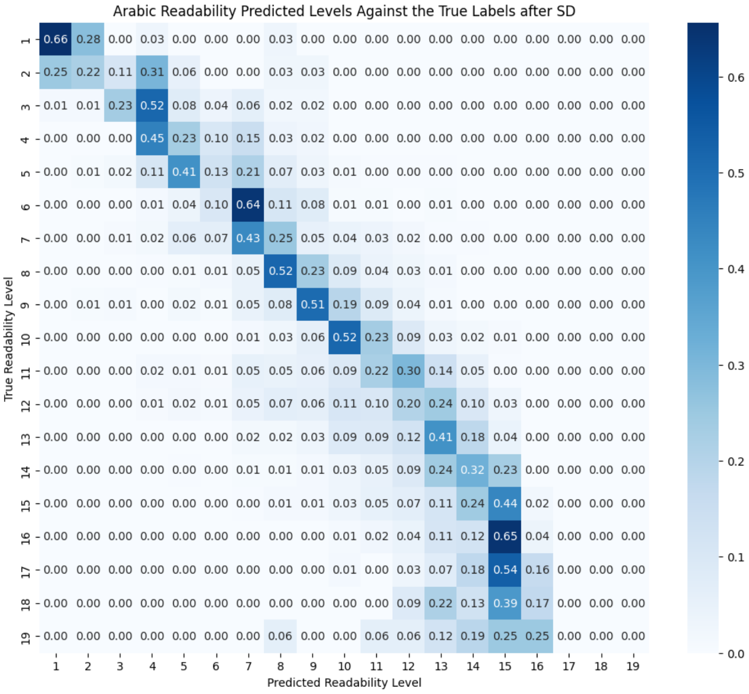


Figure 2: A normalized confusion matrix (heat map) of predicted levels against the true levels of texts in the test dataset **after** stripping diacritics.

Pixels at BAREC Shared Task 2025: Visual Arabic Readability Assessment

Ben Sapirstein
Reichman University
ben.sapirstein@post.runi.ac.il

Abstract

We present a visual-language approach to Arabic readability assessment using the PIXEL Vision Transformer, which processes rendered text as images to bypass tokenization challenges. Our system participated in the BAREC 2025 Shared Task (Sentence-level Strict track). We evaluate orthographic variants (normalization, diacritization, transliteration) and morphological segmentation with different visual boundary markers. Results show that diacritization provides useful visual cues for disambiguation, morphological segmentation improves over word-level processing, and transliterated scripts outperform native Arabic script. Our approach demonstrates the potential of visual processing for readability assessment in complex languages and writing systems.

1 Introduction

Text readability is fundamental to effective comprehension, retention, reading speed, and engagement, with texts exceeding a reader's ability often leading to disengagement and frustration (DuBay, 2004). For Arabic, a language spoken by over 400 million people worldwide, developing robust readability assessment models is crucial for advancing literacy, language learning, and academic performance (Elmadani et al., 2025b). These models are essential for educators to prepare appropriate reading materials and enhance the learning experience, making complex concepts accessible to a wide range of students across the Arab world's linguistically diverse populations. Arabic readability assessment presents significant challenges rooted in the language's morphological richness, dialectal variants, orthographic ambiguity and inconsistency (Habash, 2010), and the profound implications of these complexities on standard tokenization methods.

We introduce an alternative approach: treating text as a visual signal. By rendering Arabic sentences as images and processing them with the PIXEL Vision Transformer (Rust et al., 2022), we aim to capture readability cues directly from the graphetic and typographical properties of the text. This approach offers several advantages: (1) It bypasses the vocabulary bottleneck of token-based models, avoiding sparsity and tokenization errors; (2) It naturally encodes orthographic and morphological variation; and (3) It facilitates cross-language and cross-script transfer from large-scale pretraining.

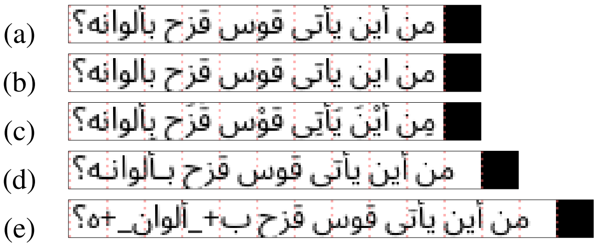


Figure 1: Visual comparison of Arabic script input variants, from top: (a) Default, (b) Normalized, (c) Diacritized, (d–e) Morphological segmentation (Tatweel and default).

We describe our submission to the BAREC 2025 Shared Task on Arabic readability assessment (Elmadani et al., 2025a), where we: (1) Apply PIXEL to sentence-level Arabic readability; (2) Compare orthographic variants including normalization, diacritization, and transliteration; (3) Evaluate morphological segmentation schemes with different visual boundary markers.

Our experiments reaffirm PIXEL's robustness on orthographic variance and reveal that diacritization provides beneficial visual disambiguation cues, morphological segmentation can improve performance, and transliterated scripts yield more tractable visual patterns. The findings highlight the potential of visual processing for readability assessment in complex languages and writing systems.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 350–356
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

2.1 Arabic Readability Assessment

The Arabic readability assessment landscape features several important datasets and frameworks. Taha-Thomure (2017) developed a 19-level text leveling framework for children's literature, adopted by the Arab Thought Foundation's Arabi21 initiative to tag over 9,000 books. This procedural framework outlines ten qualitative and quantitative criteria, including text genre, abstractness of ideas, vocabulary, text authenticity, and sentence structure, primarily targeting full texts and early education

The SAMER project contributed a five-level readability lexicon for Modern Standard Arabic (Al Khalil et al., 2020), initially containing 26,000 lemmas and later expanded to more than 40,000. The lexicon was manually annotated in triplicate by language professionals from three regions of the Arab world and with detailed annotation guidelines. SAMER also produced the first manually annotated Arabic text simplification corpus (Alhafni et al., 2024), 159K words from 15 fiction novels with document- and word-level annotations. These efforts are supported by practical applications such as the Google Docs add-on by Hazim et al. (2022), which visualizes word-level readability to assist human annotators in text simplification

Leveraging the SAMER project resources, Liberato et al. (2024) systematically explored different modeling approaches for Arabic readability assessment, ranging from rule-based methods to Arabic pretrained language models. Their research benchmarked models on a newly created corpus at both word and sentence fragment levels, highlighting the challenges posed by Arabic's morphological richness and limited readability resources. Their findings demonstrated that combining different modeling techniques yielded the best results.

Further extending these initiatives, the Balanced Arabic Readability Evaluation Corpus (BAREC) (Elmadani et al., 2025b) provides a large-scale, fine-grained dataset consisting of 1,922 documents with 69,441 sentences spanning over 1 million words. This corpus is carefully curated to cover 19 readability levels, from kindergarten to postgraduate comprehension, balancing genre diversity, topical coverage, and target audiences. BAREC is considered the largest and most fine-grained manually annotated Arabic readability resource to date (Habash et al., 2025).

2.2 Arabic Processing Challenges

Arabic poses major challenges for NLP tasks.

Morphological richness is a significant characteristic, entailing complex inflections and cliticization. Arabic words inflect for numerous grammatical features such as gender, number, person, case, aspect, mood, and voice, while also incorporating various attachable proclitics (e.g., conjunctions, prepositions, definite article) and enclitics (e.g., pronominal objects) (Liberato et al., 2024). This complexity leads to an extensive number of word forms; for example, Modern Standard Arabic (MSA) verbs alone can have upwards of 5,400 forms (Obeid et al., 2020). Such morphological complexity results in lexical sparsity and significantly complicates tasks like tokenization. In fact, Arabic exhibits a vocabulary growth rate approximately 2.5 times higher and out-of-vocabulary rates about 10 times higher than English (Habash, 2010).

Dialectal variations further complicate Arabic processing. While MSA is the formal written standard used in education, media, and literature across the Arab world, it is not the native language of any Arab speaker. Instead, native speakers communicate using a diverse array of informal spoken dialects that differ considerably from MSA and from each other in their phonology, morphology, lexicon, and even syntax (Habash, 2010). A key issue is the general lack of standardized spelling systems for Arabic dialects, which contributes to orthographic inconsistency. For instance, different forms of the letter Alif (ا ، أ ، إ ، آ) can represent the same linguistic unit: the common writing راس instead of the standard رأس results in different character codes despite conveying the same word. Orthographic normalization addresses this issue by converting letter variants or visually similar letters into a single, standardized form (Obeid et al., 2020). At the same time, informal sociolinguistic norms often guide how dialects are written, and NLP systems must be able to recognize and adapt to these conventions to fully leverage the information such texts provide.

Orthographic ambiguity is a pervasive problem in written Arabic. This means that a single written form can correspond to multiple different meanings and grammatical analyses. For example, the word درسها (drshA) can be interpreted in several ways depending on the implied diacritics: as a verb meaning 'he taught her', another verb meaning 'he studied it', or a noun phrase meaning 'her lesson'. While automatic disambiguation methods, such as Maximum Likelihood Estimation (MLE) disambiguators (Khalifa et al., 2016), attempt to resolve this issue by inserting diacritical marks that specify short vowels and consonantal geminations, the resulting proliferation of unique tokens further intensifies lexical sparsity and adds to the vocabulary bottleneck already posed by Arabic's morphological richness.

Script complexity and allographic variation pose additional challenges for visual processing. The Arabic script provides multiple different graphs that can represent the same letters as in contextual forms (e.g Ayin variants ـعـ ، عـ ، ـع ، ع), multi-character ligatures and complex word-level ligatures. While Unicode normalization can be applied to avoid inflated token vocabularies (Obeid et al., 2020), standard font features will map even Unicode-standardized input to different graphs, leading to visual variation.

Transliteration schemes such as Buckwalter (BW) and Habash-Soudi-Buckwalter (HSB) (Habash et al., 2007), offer an alternative approach to handling Arabic's orthographic complexity. HSB is particularly beneficial for visual processing, as different Arabic letter variants are mapped to visually similar Latin glyphs while preserving the orthographic distinctions of the source script (Figure 2). Additionally, Latin-based representations present fewer rendering challenges since they do not exceed typical line boundaries, unlike certain Arabic diacritics and punctuation marks.

2.3 Visual Embeddings for Language

The PIXEL model (Rust et al., 2022) treats text as images by rendering text in fixed fonts and processing image patches through Vision Transformers (Dosovitskiy et al., 2020). PIXEL is built upon the architecture of Masked Autoencoders (He et al., 2021), which are scalable self-supervised learners that use an asymmetric encoder-decoder design and masking to reconstruct missing image pixels for efficient visual representation learning. PIXEL has demonstrated strong performance as a foundation model across various languages and scripts, including Arabic, where it achieves near-parity with token-based models on core NLP tasks (95.7% vs. 95.4% POS tagging accuracy compared to BERT; 77.3 vs. 77.7 LAS in dependency parsing).

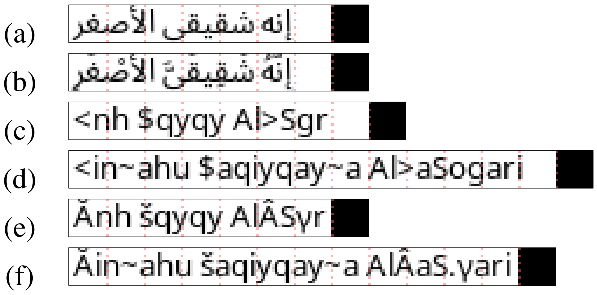

(a)
إنه شقيقي الأصغر
(b)
إِنَّهُ شَقِيقَيَّ الأَصْغَرِ
(c)
<nh $qyqy Al>Sgr
(d)
<in~ahu $aqiyqay~a Al>aSogari
(e)
Ănh šqyqy AlÂSγr
(f)
Ăin~ahu šaqiyqay~a AlÂaS.γari

Figure 2: Visual comparison of script variants before and after diacritization: (a,b) Arabic script, (c,d) Buckwalter, (e,f) HSB. Transliterated forms properly display all diacritic information, with HSB maintaining intuitive visual representations of Arabic letter variants such as different Alif forms (ا *A*, أ *Â*, إ *Ă*) .

The PIXEL model addresses some of the mentioned challenges: orthographic variations often appear as visually similar glyphs, and the visual representation allows accessing morphemes without tailored tokenization. This continuous vocabulary representation is particularly useful for dialectal data, as demonstrated by experiments on German dialects (Muñoz-Ortiz et al., 2024). However, there is still a potential pitfall when processing allographs.

This approach naturally handles RTL scripts, though Rust et al. (2022) note processing limitations where RTL sentences are processed from end to beginning, potentially affecting positional learning.

2.4 BAREC Shared Task 2025

The BAREC Shared Task 2025 focuses on fine-grained Arabic readability assessment, participants in the shared task are challenged to build models for both sentence-level and document-level readability classification.

A strong baseline for this task, as established in the research accompanying the BAREC corpus, is AraBERTv2 (Antoun et al., 2020). This model, when used with the D3Tok input variant and Cross-Entropy loss, achieved the best performance across various metrics in initial benchmarking experiments. We compare our results to the Word input variant.

3 System Overview

Our pipeline begins by rendering each Arabic text as an RGB image. We use Noto Sans Arabic at a fixed font size on a white background, following the standard PIXEL methodology (Rust et al., 2022). Sentences are rendered to a fixed image size determined by the patch size and the maximum sequence length. The image is then split into non-overlapping 16×16 patches, each patch is flattened and linearly projected to the ViT encoder. For fine-tuning, we append a linear classification head, with softmax and cross-entropy loss.

4 Experimental Setup

4.1 Text Processing Variants

We introduce two dimensions of preprocessing:

Orthographic encoding manipulates surface forms to test the effect of script and phonological cues. For Arabic script, we evaluate the individual effects of (i) dediacritization and (ii) orthographic normalization, as well as their combination, and compare them to the default and fully diacritized forms (via CAMeL's MLE disambiguator). For transliterated scripts we restrict evaluation to three variants (default, normalized+dediacritized, and diacritized).

Morphological encoding manipulates word structure. Using CAMeL Tools' MLE-based tokenizer, we segment words into stems and clitics (e.g., وكتابها → و + كتاب + ها). To make these boundaries visually salient, we experiment with different markers: standard ASCII markers (+_ and _+), Arabic tatweel to maintain script consistency, and spaces treating morphemes as distinct visual units (Figure 1).

4.2 Evaluation Metrics

We report results on Accuracy, ±1 Accuracy, MAE, and Quadratic Weighted Kappa (QWK) as the primary metric which measures agreement while accounting for the ordinal distance between predicted and true levels.

5 Results and Analysis

5.1 Orthographic Encoding Effects

Table 1 summarizes the impact of orthographic variants across Arabic, Buckwalter, and HSB scripts. A consistent pattern emerges: transliterated scripts outperform Arabic script across all metrics, with HSB achieving the highest QWK (69.3%), followed by Buckwalter (68.0%), while Arabic peaks at 66.5%. This "script gap" of approximately 3-4 QWK points suggests that visual regularity in Latin-based representations provides advantages for the vision transformer architecture.

Within each script, preserving orthographic and diacritic distinctions generally benefits PIXEL performance more than normalization. Diacritization shows particular promise for transliterated scripts, improving QWK by 1.3 points for Buckwalter and 2.4 points for HSB. However, diacritization effects in Arabic script are mixed, possibly due to incomplete visual rendering of diacritical marks that extend beyond typical line boundaries.

5.2 Morphological Encoding Effects

Table 2 presents the impact of morphological segmentation on readability assessment. Morphological segmentation using D3TOK generally improves performance over word-level processing, with both tatweel and space markers achieving 67.4% and 67.0% QWK respectively, compared to 66.3% for unsegmented text. The standard ASCII markers under-perform the baseline word-level approach. The effectiveness of space separation is particularly noteworthy, despite spaces already serving as word boundaries in the text.

5.3 Official Results

For official submission, we submitted the predictions of the default Arabic script variant. Table 3 shows that our model achieved 68.4% QWK on the blind test. However, PIXEL significantly underperformed the AraBERTv2 baseline, which achieved 76.2% QWK.

6 Conclusion and Future Work

PIXEL naturally handles orthographic variation while benefiting from morphological and phonological signals in richer text representations. English pretraining benefits from Latin script regularity, though the performance gap with token-based models suggests need for further optimization.

Script	Configuration	Accuracy	±1 Acc	MAE	QWK
Arabic	Default	40.0%	53.0%	1.74	**66.5%**
	Dediacritized	41.0%	53.7%	1.74	63.9%
	Ortho Normalized	38.8%	51.5%	1.79	65.6%
	Ortho Normalized & Dediacritized	40.0%	53.3%	1.73	64.8%
	Diacritized	41.7%	54.7%	1.70	65.8%
Buckwalter	Default	42.3%	55.7%	1.70	66.7%
	Ortho Normalized & Dediacritized	43.5%	56.1%	1.70	65.0%
	Diacritized	43.4%	56.4%	1.64	**68.0%**
HSB	Default	42.7%	55.6%	1.66	66.9%
	Ortho Normalized & Dediacritized	43.5%	56.3%	1.69	64.9%
	Diacritized	43.3%	56.7%	1.61	**69.3%**

Table 1: Orthographic encoding results on the test set.

Morphological Scheme	Boundary Marker	Accuracy	±1 Acc	MAE	QWK
WORD	–	39.0%	52.9%	1.72	66.3%
D3TOK	Default (+_/_+)	40.9%	54.0%	1.74	65.4%
	Tatweel	42.0%	54.9%	1.69	**67.4%**
	Space	42.0%	55.2%	1.69	67.0%

Table 2: Morphological encoding results on the test set.

Track	Model	Test				Blind Test			
		Acc	±1 Acc	MAE	QWK	Acc	±1 Acc	MAE	QWK
Strict	PIXEL-English	40.0%	53.0%	1.74	66.5%	41.5%	56.8%	1.6	68.4%
	AraBERTv2 (WORD)	51.1%	65.1%	1.31	76.2%				

Table 3: Strict results on Official and Blind tests vs. AraBERTv2 WORD.

The 'script gap' warrants investigation across additional scripts to determine whether effects reflect Latin-specific advantages or broader visual regularity factors. Future work could explore visual augmentations, different fonts, and document-level readability assessment.

This experiment illustrates how PIXEL can be used to assess the informative potential of specific text manipulations. Tatweel, a native Arabic elongation mark, is presented here merely as an example of a script-internal feature that could be evaluated in this way, with potential relevance for human readers.

Future work could explore PIXEL's ability to capture purely visual cues that affect human reading, such as glyph similarity or diacritic placement. In particular, experiments predicting reading speed could further investigate these effects.

7 Limitations

We tested orthographic variation over the English pretrained PIXEL-base model, giving advantage for Latin characters over Arabic.

We used the default render configuration and it occasionally rendered Arabic script outside of the image.

Acknowledgments

We thank the BAREC shared task organizers for providing the dataset and evaluation framework. We also acknowledge the computational resources provided by our institution's computing cluster.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

William H DuBay. 2004. The principles of readability. *Online submission*.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Nizar Habash, Abdelhadi Soudi, and Timothy Buckwalter. 2007. On Arabic Transliteration. In Abdelhadi Soudi, Antal Van Den Bosch, and Günter Neumann, editors, *Arabic Computational Morphology*, volume 38, pages 15–22. Springer Netherlands, Dordrecht. Series Title: Text, Speech and Language Technology.

Nizar Habash, Hanada Taha-Thomure, Khalid Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2021. Masked autoencoders are scalable vision learners. *Preprint*, arXiv:2111.06377.

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2016. Yamama: Yet another multi-dialect arabic morphological analyzer. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations*, pages 223–227.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Alberto Muñoz-Ortiz, Verena Blaschke, and Barbara Plank. 2024. Evaluating pixel language models on non-standardized languages. *arXiv preprint arXiv:2412.09084*.

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.

Phillip Rust, Jonas F Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2022. Language modelling with pixels. *arXiv preprint arXiv:2207.06991*.

Hanada Taha-Thomure. 2017. *Arabic Language Text Leveling* (معايير هنادا طه لتصنيف مستويات النصوص العربية). Educational Book House (دار الكتاب التربوي للنشر والتوزيع).

A Additional Experimental Details

All models were trained using PyTorch 2.5.1 with CUDA 12.4 on two NVIDIA GeForce RTX 3090 GPUs. The rendering pipeline used PangoCairo text renderer. We preprocess all variants with Unicode-normalization and tatweel removal. We used the default architecture composed of 12 Transformer layers, hidden size of 768, 12 attention heads, totaling 86M encoder parameters.

A.1 Training Hyperparameters

Fine-tuning: 86M parameters, sequence length 256, batch size 64, learning rate 5e-05, 7 epochs, dropout 0.1, model selection based on Dev set Cross Entropy loss. Morphological encoding variants were trained on half the batch size and learning rate and on a single GPU.

Code Availability

Our code is available at: `https://github.com/bensapirstein/pixel`

Phantoms at BAREC Shared Task 2025: Enhancing Arabic Readability Prediction with Hybrid BERT and Linguistic Features.

Ahmed Alhassan
Carnegie Mellon University Africa
aalhassa@andrew.cmu.edu

Asim Mohamed
African Institute for Mathematical Sciences
amohamed@aimsammi.org

Moayad Elamin
Carnegie Mellon University Africa
melamin@alumni.cmu.edu

Abstract

This paper describes our system for the BAREC 2025 Shared Task on Arabic Readability Assessment. Our approach is centered on a hybrid model that combines the deep contextual representations of a pre-trained transformer (AraBERTv02) with a rich set of engineered linguistic features. We extracted over 200 lexical, morphological, syntactic, and semantic features, which were refined to the 100 most informative ones through a multi-stage selection process. Our final model demonstrates significant effectiveness, achieving a **Quadratic Weighted Kappa (QWK) of 82.7%** and an exact accuracy of **57.6%** on the official blind test set. These results highlight the powerful synergy between transformer-based embeddings and explicit linguistic signals for the nuanced task of assessing Arabic text readability.

1 Introduction

Automatic Readability Assessment (ARA) aims to predict the difficulty level of a given text for a target audience. Although extensively studied for English, ARA for Arabic remains a developing field, presenting unique and significant challenges for modern Natural Language Processing (NLP) models (Liberato et al., 2024). The complexity of Arabic, which comes from its rich derivational morphology, optional diacritization, and the widespread phenomenon of diglossia, complicates the extraction of reliable readability features. Traditional readability formulas, often translated into English, do not capture these linguistic nuances. More recent machine learning and deep learning models have shown promise (Hazim et al., 2022), yet their performance is often constrained by the scarcity of large, high-quality, and fine-grained annotated corpora for Arabic.

The BAREC Shared Task 2025 on Arabic Readability Assessment (Elmadani et al., 2025a) directly addresses this gap by introducing a new, large-scale, and balanced corpus designed for this purpose (Elmadani et al., 2025b) . This initiative provides a crucial benchmark for the development and evaluation of sophisticated Arabic ARA systems. The task challenges participants to move beyond surface-level features and explore more complex linguistic and semantic representations to accurately predict readability scores.

In this paper, we present our system for the BAREC Shared Task. Our approach is novel in its hybrid architecture, which synergistically combines deep contextual embeddings from a pre-trained Arabic transformer model with a rich set of handcrafted linguistic features. These features are specifically designed to capture the morphological, syntactic, and psycholinguistic dimensions of Arabic text that influence reading comprehension. By integrating these diverse feature sets, our model aims to create a more holistic and accurate representation of text complexity. We hypothesize that this multi-faceted approach will outperform models that rely solely on either deep learning or traditional feature engineering, thereby setting a new standard for Arabic readability assessment.

2 Background

The BAREC Shared Task 2025 (Elmadani et al., 2025a) focuses on fine-grained, sentence-level readability assessment for Modern Standard Arabic. The primary goal is to predict a readability score for a given Arabic sentence on a continuous scale. The task is structured into three main tracks:

- **Open Track:** Participants are allowed to use any external data, resources, or pre-trained models to build their systems.
- **Constrained Track:** Participants are restricted to using only the provided training set of BAREC Corpus (Elmadani et al., 2025b) and specific, pre-approved external resources,

Proceedings of The Third Arabic Natural Language Processing Conference, pages 357–361
November 8-9, 2025 ©2025 Association for Computational Linguistics

namely the SAMER Corpus (Alhafni et al., 2025) and the SAMER Lexicon (Al Khalil et al., 2020).

- **strict Track:** Participants are restricted to using only the provided training set of BAREC Corpus.(Elmadani et al., 2025b)

We participated in the **strict track**.

The task utilizes BAREC (Balanced Arabic Readability Corpus) (Elmadani et al., 2025b), a comprehensive dataset containing sentences sourced from diverse genres and annotated according to detailed guidelines (Habash et al., 2025). Each sentence in the corpus is assigned a readability score derived from expert human annotations, which reflects the cognitive effort required for a reader to understand it. An example of an input sentence and its corresponding output score is shown below:

Input: بين طعن القَنا وخَفْق البُنودِ
(Translation: Between the thrust of spears and the fluttering of banners.)
Output: 17

Prior work in Arabic readability has evolved significantly. Early studies focused on adapting the classic readability formula, such as the Flesch-Kincaid index, which mainly uses shallow features such as word and sentence length. Later research incorporated more sophisticated and Arabic-specific linguistic features, including morphological complexity and syntactic structures, into machine learning frameworks like Support Vector Machines (SVM) and Random Forests (Cortes and Vapnik, 1995) (Breiman, 2001). With the advent of deep learning, researchers began to leverage neural networks and, more recently, large pre-trained language models like AraBERT (Antoun et al., 2020) and CAMeLBERT (Inoue et al., 2021). These models have demonstrated strong performance by learning rich semantic representations directly from text. Our work builds upon these advances by proposing a hybrid system that leverages the strengths of both feature-based and deep learning paradigms, a strategy we believe is crucial for capturing the multifaceted nature of text readability in Arabic.

3 System Overview

Our system is designed to address the multifaceted challenge of Arabic text readability by integrating deep contextual understanding with explicit linguistic knowledge. The core of our approach is a hybrid neural architecture that leverages a pre-trained transformer model alongside a curated set of engineered features.

Design Rationale: The primary challenge in readability assessment is to capture a wide range of signals, from syntactic complexity and lexical choice to semantic coherence. While pre-trained models like BERT excel at learning contextual representations, they may not explicitly capture specific linguistic phenomena known to influence readability. Our design decision to fuse BERT with handcrafted features is motivated by this; we provide the model with both implicit, learned representations and explicit, targeted linguistic cues, creating a more robust and informed system.

Algorithmic Framework: Our model, implemented in PyTorch and the Hugging Face `transformers` library, consists of two main components: a text encoding module and a feature fusion classifier.

1. **Textual Representation:** We use the `aubmindlab/bert-base-arabertv02` model to generate contextualized embeddings for the input text. For a given sentence, the final hidden state of the special `[CLS]` token is used as its aggregate semantic representation. Let this be denoted as $\mathbf{e}_{\text{text}} \in \mathcal{R}^{768}$.

2. **Linguistic Feature Representation:** The 100 features selected from our feature engineering pipeline are compiled into a numerical vector, $\mathbf{f}_{\text{raw}}$. This vector is standardized using a `StandardScaler` (fit on the training data) to ensure zero mean and unit variance, resulting in the final feature vector $\mathbf{f}_{\text{num}}$.

3. **Hybrid Feature Fusion:** The textual and linguistic representations are combined through concatenation to form a unified feature vector:

$$\mathbf{c} = [\mathbf{e}_{\text{text}} \oplus \mathbf{f}_{\text{num}}]$$

   where $\oplus$ denotes the concatenation operation. This vector $\mathbf{c} \in \mathcal{R}^{768+100}$ serves as input to the final classification layer.

4. **Classification Head:** The combined vector $\mathbf{c}$ is passed through a multi-layer perceptron (MLP) to predict the readability level. This layer is trained to classify the input into one of the 19 ordinal readability classes.

Training Configuration: The model is trained for a maximum of 10 epochs using the AdamW optimizer with a learning rate of 5×10^{-5} and a linear learning rate scheduler. To counteract class imbalance, we employ a weighted Cross-Entropy Loss, with weights inversely proportional to class frequencies. We utilize mixed-precision training for efficiency. The model's performance is monitored on the validation set using the Quadratic Weighted Kappa (QWK) score, and we apply early stopping with patience of 2 epochs to prevent overfitting.

4 Experimental Setup

Dataset: We utilized the BAREC sentence-level dataset for our experiments. The data is partitioned into three distinct sets: a training set for model development, a validation set for hyperparameter tuning, and a test set for final evaluation. The respective sizes and characteristics of these splits are determined by the original dataset providers. In addition to the standard splits, we also process the sentence-level blind test set.

Preprocessing and Feature Engineering: To prepare the data for our models, we implement a comprehensive pre-processing and feature engineering pipeline.

Text Normalization: Each sentence undergoes a series of normalization steps using the `camel-tools` library(Obeid et al., 2020). This includes Unicode normalization, normalization of Alef (أ, إ, آ to ا), Alef Maksura (ى to ي), and Teh Marbuta (ة to ه), followed by the elimination of all diacritics.

Feature Extraction: We extract a rich set of more than 200 features from the normalized text, leveraging the capabilities of `camel-tools`. These features can be categorized as follows:

- **Surface Features:** Basic statistics such as word count, average and standard deviation of word length, and the ratio of long ($>=$ 7 characters) and short ($<=$ 3 characters) words.
- **Character-level Features:** Ratios of non-Arabic characters, punctuation, numbers, mathematical operators, and other symbols within each sentence.
- **Morphological Features:** Proportions of various parts of speech (POS), gender, number, aspect, case, and other morphological characteristics derived from the top analysis of an MLE disambiguator. We also compute morphological richness, verb-to-noun ratio, and affix ratios (prefix, suffix) based on morphological tokenization.
- **Semantic Features:** We include the count and ratio of Named Entities (NER), a sentiment score (positive, neutral, negative) and dialect identification scores, particularly the confidence score for Modern Standard Arabic (MSA).
- **Lexical Features:** The ratio of stop words in a sentence and the stem diversity, calculated as the ratio of unique stems to the total number of stems.

Feature Selection: To reduce dimensionality and mitigate multicollinearity, we apply a three-stage feature selection process to the training data:

1. **Variance Thresholding:** Features with variance below a threshold of 0.01 are removed.
2. **Correlation Filtering:** Highly correlated features are filtered out. We compute the Pearson correlation matrix and remove one feature from any pair with a correlation coefficient greater than 0.95.
3. **Tree-based Selection:** A Random Forest classifier is trained on the remaining features to rank their importance. The top 100 most informative features are selected for the final feature set.

Implementation Details

Our primary model is a hybrid architecture that combines a pre-trained transformer with the engineered numerical features. The model is built using PyTorch and the Hugging Face `transformers` library.

Model Architecture: We use the `aubmindlab/bert-base-arabertv02` model as our text encoder. The output representation of the `[CLS]` token is extracted and concatenated with the vector of scaled numerical features. This combined vector is then passed through a classification head consisting of a linear layer, a SiLU activation function, a dropout layer ($p = 0.2$), and a final linear layer to produce the output logits for the 19 readability classes. A dropout layer ($p = 0.3$) is also applied to the combined feature vector before it enters the classifier.

Training: The model is trained for a maximum of 10 epochs with a batch size of 16. We use the AdamW optimizer with a learning rate of 5×10^{-5} and a linear learning rate scheduler. To address class imbalance, we employ a weighted Cross-Entropy Loss function, where weights are inversely proportional to class frequencies in the training set. We utilize mixed-precision training to accelerate computation. Early stopping is implemented with a patience of 2 epochs, monitored by the validation Quadratic Weighted Kappa (QWK) score. The best-performing model based on validation QWK is saved for evaluation.

Evaluation Metrics

Given the ordinal nature of the readability labels, we evaluated model performance using a suite of metrics. In addition to standard classification and regression metrics like **Exact Accuracy** and **Mean Absolute Error (MAE)**. We also report **Adjacent Accuracy** (allowing for an off-by-one error), **the 3, 5, and 7 Levels Accuracy**—classifying the sentences as if they are classified into 3, 5, and 7 different classes, respectively—and the **Quadratic Weighted Kappa (QWK)**, which is particularly well-suited for measuring inter-rater agreement on an ordinal scale.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

$$\text{QWK} = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}}$$

where w_{ij} are the weights, O_{ij} is the observed count, and E_{ij} is the expected count for a label pair (i, j).

5 Results

Our system's performance was evaluated on the official BAREC blind test set. We also conducted internal experiments to compare different configurations of our model's classification head on the development set. The internal comparison results are included in Appendix A. The evaluation focuses on metrics suited for ordinal classification, primarily **Quadratic Weighted Kappa (QWK)**, alongside **Exact Accuracy**, **Adjacent Accuracy (Acc ±1)**, and **Mean Absolute Error (MAE)**.

5.1 Official Blind Test Set Results

On the official competition blind test set, our final model achieved a strong performance, demonstrating its robustness and generalization capabilities. The system attained a **QWK of 82.7%**, confirming a high level of agreement with the gold-standard labels. The exact accuracy was **57.6%**, while the adjacent accuracy (Acc ±1) reached **72.3%**, indicating that most of our model's errors were minor, differing by only a single readability level. The complete results are presented in Table 1.

QWK	Acc	Acc ±1	MAE
82.7%	**57.6%**	**72.3%**	**1.06**

Acc (3)	Acc (5)	Acc (7)
77.2%	71.3%	67.4%

Table 1: Final results of our system on the official sentence-level blind test set.

The high QWK and adjacent accuracy scores validate our hybrid approach, confirming that combining pre-trained language models with carefully engineered linguistic features is highly effective for sentence-level readability assessment in Arabic.

6 Conclusion

In this paper, we present our system for the BAREC 2025 Shared Task on sentence-level Arabic Readability Assessment. Our approach successfully integrated a powerful pre-trained Arabic transformer model with a comprehensive set of linguistic features to create a robust prediction system. The final model achieved an impressive **Quadratic Weighted Kappa of 82.7%** on the blind test set, demonstrating the efficacy of our methodology.

Our key finding is that, while transformers are excellent at capturing semantic context, their performance is significantly enhanced by explicit features that describe lexical complexity, morphological richness, and sentence structure. This hybrid strategy proved crucial for navigating the subtleties of the Arabic language. Future work could involve exploring more advanced transformer architectures, incorporating features from diverse linguistic resources, and conducting a thorough error analysis to better understand the remaining challenges in automatic readability assessment.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2025. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. 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.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. 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.

A Internal Model Comparison

To select the best architecture, we compared three variants of our BERT-based model on the development set: one using a SiLU activation function, one using the Swish function, and one employing an ordinal regression head. The results, summarized in Tables 2 3, show that the models with **SiLU** and **Swish** activation functions performed very similarly and slightly better than the ordinal regression approach across most metrics. Based on its marginally higher QWK score, the BERT (SiLU) configuration was selected for the final submission.

Model	Accuracy	Accuracy ±1	MAE
BERT (swish)	56.18%	70.66%	1.0917
BERT (SiLU)	55.87%	69.90%	1.1023
BERT (ordinal)	53.61%	69.78%	1.1473

Table 2: Model Performance Metrics (Part 1)

Model	QWK	Acc (7)	Acc (5)	Acc (3)
BERT (swish)	81.15%	65.45%	69.01%	74.60%
BERT (SiLU)	81.17%	64.41%	67.95%	74.55%
BERT (ordinal)	79.35%	63.38%	67.06%	72.87%

Table 3: Model Performance Metrics (Part 2)

STBW at BAREC Shared Task 2025: AraBERT-v2 with MSE-SoftQWK Loss for Sentence-Level Arabic Readability

Saoussan Trigui
Independent Researcher
triguisaoussan51@gmail.com

Abstract

Automatic Readability Assessment estimates how hard a text is for its target readers, using features such as vocabulary, spelling, morphology, etc. Based on this premise, we evaluate our experiments on Arabic language under the BAREC 2025 shared task protocol. This paper addresses the sentence-level readability assessment task with strict track, that allows only the use of BAREC train set to predict Arabic readability on a fine-grained 19-level scale. Our solution is based on a two-phase fine-tuning of AraBERT-v2 on a custom feature set of the BAREC corpus. In the blind test set, the system achieves a QWK of 85.6%.

1 Introduction

Automatic Readability Assessment (ARA) is the task of computationally modeling the reading and comprehension difficulty of a text for a specific target audience. Its applications are diverse and impactful, spanning human-facing scenarios such as selecting appropriate educational materials for language learners, supporting readers with learning disabilities, and facilitating self-directed learning. In machine-facing contexts, ARA is instrumental in ranking search results by complexity, controlling the reading level of machine-translated output, and evaluating the efficacy of automatic text simplification systems (Vajjala, 2021). For Arabic, ARA is particularly challenging due to rich morphology, orthographic variation (e.g., diacritics and normalization) and dialectal/code-switching phenomena.

In this work, we aim to build a strong yet simple Arabic ARA system for the BAREC Shared Task (Elmadani et al., 2025a). We participate in the strict track of the sentence-level readability assessment where models must be trained exclusively on the training set of the Balanced Arabic Readability Evaluation Corpus (BAREC)[1] (Habash et al., 2025).

We cast sentence readability as scalar regression and then explicitly align training with the evaluation metric Quadratic Weighted Kappa (QWK). Concretely, we fine-tune AraBERT in two phases: an MSE warm-up followed by a differentiable QWK objective (SoftQWKLoss) that converts the scalar prediction into soft, distance-aware probabilities over the 19 levels and optimizes $(1 - \mathrm{QWK})$ on a soft confusion matrix. On the input side, we inject text metadata and statistics as well as linguistic cues (D3Tok) derived from CAMeL tools (Obeid et al., 2020). Empirically, this combination reaches QWK test results of 84.88, improving slightly over the 83.9 QWK score yielded by MSE-only phase. On the blind test leaderboard, we ranked in the 4th position out of 16 participations, with a 85.6 QWK for the sentence readability level subtask. The code of this solution is publicly available[2].

In summary, the proposed solution is composed of: (1) a compact AraBERT pipeline for Arabic readability that requires minimal feature engineering yet remains competitive, and (2) a metric-aligned training recipe (MSE → SoftQWK) that is architecture-agnostic and easy to reproduce.

Next, we present some background and we formalize the task and its input/output setup; then we present our method and training objectives, describe the experimental setup, and report results. We follow with error analyses, discuss limitations, and conclude.

2 Background

2.1 History of Automatic Readability Assessment (ARA)

The origins of ARA date back nearly a century to the development of manually computed readability formulas. These formulas are characteristically simple, often expressed as weighted linear func-

[1] https://barec.camel-lab.com/sharedtask2025

[2] https://github.com/Saoussan/BAREC_Arabic_Readability_Assessment

Proceedings of The Third Arabic Natural Language Processing Conference, pages 362–366
November 8-9, 2025 ©2025 Association for Computational Linguistics

tions of easily quantifiable, surface-level textual features (Vajjala, 2021). Among the most influential and enduring of these is the Flesch Reading Ease formula (Flesch, 1948), which calculates a score on a 0-100 scale, based on average sentence length and average word length in syllables, and the Dale-Chall formula which uses a predefined list of common words to identify "difficult" vocabulary (Dale and Chall, 1948).The shift towards supervised machine learning, which reframes readability assessment as a classification or regression problem, allowed for the integration of richer sets of linguistic features. Algorithms like Support Vector Machines (SVM) or Random Forests, demonstrated superior performance compared to traditional formulas (Imperial and Kochmar, 2023). While researchers have progressed to complex neural network architectures, the traditional, simpler formulas continue to exist, especially in fields like education or healthcare, that value prediction interpretability (Vajjala, 2021).

2.2 Challenges of Arabic Language

Applying ARA methodologies to the Arabic language presents a set of challenges that are not adequately addressed by models developed primarily for English. These challenges stem from the inherent linguistic characteristics of Arabic, which impact text complexity (Cavalli-Sforza et al., 2018). First, Arabic is a morphologically rich language, characterized by a highly inflectional system. This means that surface-level metrics like average word length in characters, may be poor indicators of difficulty for Arabic. Second, Arabic orthography is marked by an ambiguity due to the optionality of diacritics (short vowel markings) in most written texts. A single undiacritized word form can correspond to multiple distinct words with different meanings and pronunciations, which can only be resolved through context. Third, no one speaks the Modern Standard Arabic (MSA) as a native mother tongue, a language that can differ substantially in lexicon, phonology, and grammar from the daily dialect. This complicates the very definition of a "target reader" and may make the task of assessing readability for L1 speakers challenging (Cavalli-Sforza et al., 2018).

2.3 ARA for Arabic

The research community has recently focused on Arabic text readability providing scientific resources (Al Khalil et al., 2020; Alhafni et al., 2024; Elmadani et al., 2025b; Habash et al., 2025; Hazim et al., 2022). The trajectory of Arabic ARA has largely mirrored that of English, beginning with attempts to adapt or create formulas tailored for Arabic (El-Haj and Rayson, 2016; Cavalli-Sforza et al., 2018; Liberato et al., 2024) and machine learning techniques (Cavalli-Sforza et al., 2018; Bessou and Chenni, 2021). Recently, we witnessed the development of pre-trained language models (PLMs), pre-trained specifically for the Arabic language (Inoue et al., 2021; Liberato et al., 2024; Antoun et al., 2020). Upon its release, AraBERT established new state-of-the-art results across various Arabic NLP benchmarks. In this work, we propose a two-phase fine-tuning of AraBERT PLM to predict Arabic text readability levels.

3 System Overview

For our experiments, we build upon the AraBERT-v2 baseline (Elmadani et al., 2025b), but extend it with additional features and a two-phase optimization strategy. Specifically, we fine tune AraBERT-v2 (Antoun et al., 2020) as the backbone encoder, while enriching its input with surface-level statistical indicators (e.g., word count and word-length statistics) and a morphologically segmented representation generated using the D3tok segmenter (Obeid et al., 2020), alongside the raw text. On top of the encoder, we employ a single-neuron regression head to predict continuous readability scores. In phase one, we fine-tune the model using mean squared error (MSE) loss to capture the ordinal nature of readability classes. Since the main evaluation metric in the challenge is QWK, we continue the fine-tuning of the model in phase two, with a differentiable Soft Quadratic Weighted Kappa (SoftQWK) loss (de la Torre et al., 2018), directly aligning optimization with the official evaluation metric (Cohen, 1968).

For this purpose, we take inspiration from (Diaz and Marathe, 2019) and turn the scalar prediction into a soft class distribution to be compatible with the SoftQWK loss. We clamp the real prediction value to a [1...19] vector and spread its mass over the 19 readability levels with a Gaussian window centered at the predicted class. That yields a soft label probability vector $P \in \mathbb{R}^K$. With one-hot gold labels T, we construct a soft confusion matrix

$$O = T^t P \qquad (1)$$

and the chance agreement E. Using the standard

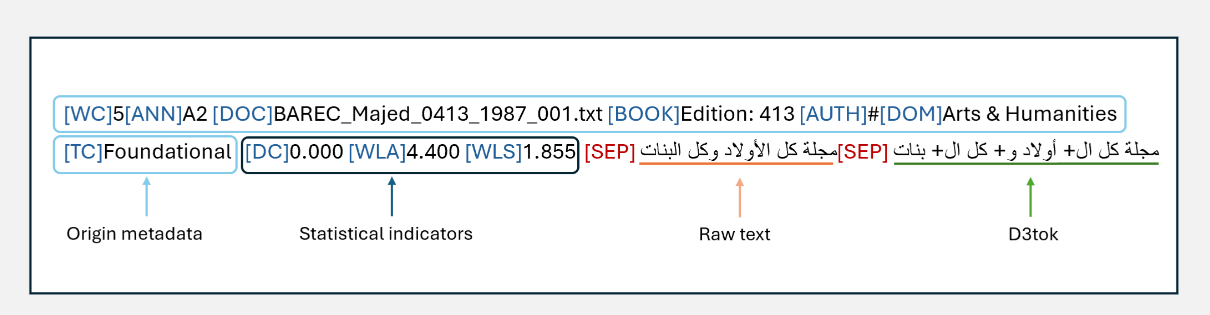

[WC]5[ANN]A2 [DOC]BAREC_Majed_0413_1987_001.txt [BOOK]Edition: 413 [AUTH]#[DOM]Arts & Humanities
[TC]Foundational [DC]0.000 [WLA]4.400 [WLS]1.855 [SEP] مجلة كل الأولاد وكل البنات [SEP] مجلة كل ال+ أولاد و+ كل ال+ بنات
Origin metadata
Statistical indicators
Raw text
D3tok

Figure 1: Example of an enriched data sample

quadratic weight matrix W, the QWK is defined as

$$\kappa = 1 - \frac{\langle W, O \rangle}{\langle W, E \rangle}, \quad (2)$$

During the training, we optimize the loss L

$$\mathcal{L} = 1 - \kappa. \quad (3)$$

4 Experimental Setup

4.1 BAREC Corpus

All our experiments are running on the Balanced Arabic Readability Evaluation Corpus (BAREC) dataset (Elmadani et al., 2025b; Habash et al., 2025). BAREC comprises over 69K sentences (around 1M words) covering three domains: Humanities, Social Sciences, and STEM and aimed at three readership groups (Foundational, Advanced, Specialized). Each sentence is annotated for readability on a fine-grained 19-level scale using guidelines developed by the authors. BAREC is considered as the largest Arabic corpus for readability assessment.

The authors establish baseline readability models (Elmadani et al., 2025b), at multiple granularities (19, 7, 5, 3 levels). We use the existing split of the BAREC Corpus which is ≈80% for training, ≈10% for validation, and ≈10% for testing (see Table 2).

Regarding feature preparation, we extend the input representation with additional components beyond the BAREC baseline setup. Each training instance is formatted as a single sequence that concatenates (i) origin metadata provided by the corpus, (ii) surface-level statistical indicators such as word count and word-length statistics, (iii) the raw text, and (iv) its D3tok-based morphological segmentation. The adopted features are listed bellow:

- **Word count**: the number of words in the raw sentence
- **Document**: the name of the source document
- **Book**: the name of the document's book
- **Author**: the name of the document's author
- **Domain**: the document's domain (one of *Arts & Humanities*, *STEM* or *Social Sciences*)
- **Text class**: the document's readership group (one of *Foundational*, *Advanced*, or *Specialized*)
- **Diacritics coverage**: frequency of diacritics in the raw sentence
- **Average word length**: the mean number of characters per word in the raw sentence
- **Word length standard deviation**: the standard deviation of the number of characters per word in the raw sentence
- **Sentence**: the raw text
- **D3tok**: morphologically segmented representation of the sentence

To ensure the model can differentiate between these heterogeneous sources of information, the components were separated by the special delimiter token `[SEP]`. We add a list of field separators as special tokens to the tokenizer (`[WC]`, `[ANN]`, `[DOC]`, `[BOOK]`, `[AUTH]`, `[DOM]`, `[TC]`, `[DC]`, `[WLA]`, `[WLS]`) in order to prevent them from being broken into subwords. This enriched representation provides the encoder with both shallow statistical cues and deeper morphological structure, while maintaining a structured and learnable input format. The model's input will look like the example in Figure 1.

4.2 Our experiments

We treat readability level prediction as a regression problem. We use a two-phase training schedule with distinct losses. In **Phase 1**, We fine-tune the AraBERT-v2 pretrained model in mixed-precision mode for 6 epochs with a batch size of 64. An

Loss	**Acc^{19}**	**$\pm$1 Acc^{19}**	**Acc^{7}**	**Acc^{5}**	**Acc^{3}**	**QWK**	**Dist**
SoftQWK	34.8%	74.0%	64.8%	72.0%	86.6%	**84.8%**	1.19
Baseline	43.1%	73.1%	61.1%	67.8%	75.9%	84.0%	1.13

Table 1: Results of our system compared to the baseline on the shared BAREC Test set

	#Documents	**#Sentences**	**#Words**
Train	1,518 (79%)	54,845 (79%)	832,743 (80%)
Dev	194 (10%)	7,310 (11%)	101,364 (10%)
Test	210 (11%)	7,286 (10%)	105,264 (10%)
All	**1,922 (100%)**	**69,441 (100%)**	**1,039,371 (100%)**

Table 2: BAREC splits

AdamW optimizer minimizes the Mean Squared Error (MSE) between the scalar prediction $\hat{y}$ and the gold label y using a learning rate of 2×10^{-5} with linear warm-up over 10% of the total updates. We consider the best checkpint on validation set, which is the third epoch, then, in **Phase 2**, we switch to the differentiable QWK objective (SoftQWKLoss): each $\hat{y}$ is converted to a Gaussian-smoothed distribution over the 19 levels, a soft confusion matrix is accumulated, and we minimize $1 - \kappa$ so that optimization is aligned with the leaderboard metric.

For evaluation, from the raw scalar $\hat{y}$ we report *MAE*. For ordinal metrics, we round and clip $\hat{y}$ to the range $[1, 19]$ and compute QWK, tolerance-1 accuracy (AdjAcc19), exact 19-way accuracy (Acc19), and coarse-bin accuracies (Acc7/Acc5/Acc3) obtained by collapsing the 19 levels into 7/5/3 groups (Elmadani et al., 2025b).

5 Results

In **Phase 1:**. We train an AraBERTv2-based system on inputs combining origin metadata, statistical indicators, raw text, and D3Tok features, using an MSE loss. This yields an evaluation QWK of 83.9. In **Phase 2:** We then fine-tune the Phase-1 model with the SoftQWK loss, reaching almost a QWK of 84.9 which is above the shared task baseline (Table 1).

Error analysis: The per-level MAE plot (Figure 2) shows the largest errors at the highest readability levels. To understand the origins of these errors, we analyse the confusion matrix (Figure 3) which indicates that many true level-18/19 items are predicted as level 16. This is clearly due to the class imbalance that affects the boundary at the top of the scale. In the future, we will investigate employing data and loss weighting techniques to tackle this problem.

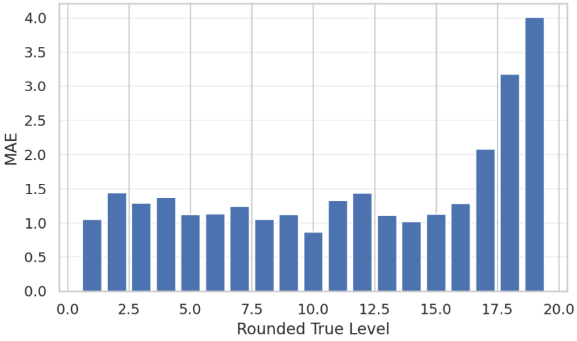


Figure 2: Per-level evaluation MAE vs. true level

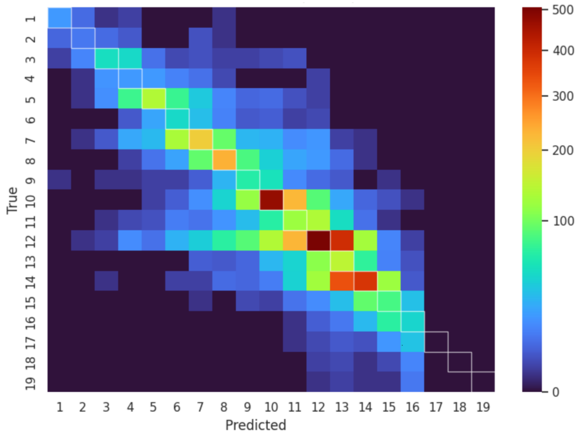


Figure 3: Confusion matrix (evaluation) of predictions cluster along the diagonal, with underestimation at high true levels (17–19).

6 Conclusion

In this work, we presented a competitive system for Arabic readability in the BAREC shared task. Using AraBERTv2 with lightweight metadata/statistics and CAMeL-derived D3Tok features from the BAREC dataset, we trained in two phases: an MSE warm-start followed by a metric-aligned SoftQWK loss. This increased QWK from 84.0% to 84.88% on the test set. Error analysis shows that most remaining mistakes occur at the highest levels (17–19), likely due to class imbalance. Going forward, we plan to mitigate this by augmenting training with the SAMER dataset (Alhafni et al., 2024) and other related resources.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

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.

Sadik Bessou and Ghozlane Chenni. 2021. Efficient measuring of readability to improve documents accessibility for arabic language learners. *arXiv preprint arXiv:2109.08648*.

Violetta Cavalli-Sforza, Hind Saddiki, and Naoual Nassiri. 2018. Arabic readability research: current state and future directions. *Procedia computer science*, 142:38–49.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Jordi de la Torre, Domenec Puig, and Aida Valls. 2018. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognition Letters*, 105:144–154. Machine Learning and Applications in Artificial Intelligence.

Raul Diaz and Amit Marathe. 2019. Soft labels for ordinal regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mahmoud El-Haj and Paul Rayson. 2016. Osman—a novel arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Joseph Marvin Imperial and Ekaterina Kochmar. 2023. Automatic readability assessment for closely related languages. *arXiv preprint arXiv:2305.13478*.

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.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. 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.

Sowmya Vajjala. 2021. Trends, limitations and open challenges in automatic readability assessment research. *arXiv preprint arXiv:2105.00973*.

LIS at BAREC Shared Task 2025: Multi-Scale Curriculum Learning for Arabic Sentence-Level Readability Assessment Using Pre-trained Language Models

Anya Amel Nait djoudi, Patrice Bellot, Adrian-Gabriel Chifu
Aix-Marseille Université, CNRS, LIS
{anya-amel.NAIT-DJOUDI, patrice.bellot, adrian.chifu}@univ-amu.fr

Abstract

Sentence-level readability assessment, which measures how easily individual sentences can be understood, has seen significant advances in English. However, Arabic readability assessment remains underexplored, primarily due to the language's morphological complexity and the scarcity of fine-grained annotated datasets. To address this gap, we leveraged the BAREC corpus, which provides 69K sentences annotated across 19 readability levels, enabling us to develop and compare five different modeling strategies ranging from lightweight classifiers to fine-tuned Arabic language models. Our experiments revealed that task-specific pretraining with CamelBERT yielded substantial performance gains, while curriculum learning offered benefits in specific scenarios. Ultimately, direct fine-tuning achieved state-of-the-art performance (QWK = 82.4). Through detailed error analysis, we identified that models struggled most with distinguishing between the lower readability level 2 and higher readability levels (15-19), highlighting the inherent challenges in fine-grained Arabic readability modeling across the full spectrum of proficiency levels.

1 Introduction

Readability assessment measures how easily a text can be read and understood by its target audience. It is crucial in various contexts including pedagogical settings, foreign language learning, health literacy (Djoudi et al., 2025), and content accessibility (Xia et al., 2016; Vajjala and Meurers, 2012; Collins-Thompson and Callan, 2004; Fox and Duggan, 2013). To enable automatic assessment, many resources have been made available ranging from datasets with annotated readability levels to readability assessment models. While progress in English readability assessment has been extensive (Azpiazu and Pera, 2019; Deutsch et al., 2020; Qiu et al., 2021; Devlin et al., 2019), Arabic readability assessment remains less studied. Arabic presents unique challenges due to its morphological richness, complex derivational patterns, and limited availability of fine-grained annotated resources. The available datasets predominantly employ binary (Soliman and Familiar, 2024) or ternary classification schemes (Al-Khalifa and Al-Ajlan, 2010), which train models with an oversimplified view of reading proficiency. Fine-grained readability levels offer the potential to better capture the continuous spectrum of literacy levels across diverse readers and provide more nuanced assessments that align with real-world reading abilities.

To address these limitations, we utilize the newly introduced BAREC corpus (Balanced Arabic Readability Evaluation Corpus) [1] (Elmadani et al., 2025b), a large-scale, fine-grained corpus containing over 69,000 sentences from 1,922 documents. The corpus spans 19 readability levels, from kindergarten (1) to postgraduate (19), covering diverse genres and domains (Arts & Humanities, Social Sciences, STEM) across three readership groups (Foundational, Advanced, Specialized).

Our contributions include systematic evaluation of five distinct model architectures, spanning lightweight MLP classifiers over pre-trained embeddings to full progressive and direct fine-tuning of Arabic language models. We demonstrate that:

1. task-specific pretraining is essential, with readability-focused CamelBERT substantially outperforming general-purpose models;

2. curriculum learning provides situational benefits in fine-tuning settings;

3. direct fine-tuning achieves state-of-the-art performance (QWK = 82.4);

4. comprehensive error analysis reveals difficulty in distinguishing lower readability level 2 and higher readability levels (15 to 19).

[1] BAREC corpus: `https://huggingface.co/datasets/CAMeL-Lab/BAREC-Shared-Task-2025-sent`

Proceedings of The Third Arabic Natural Language Processing Conference, pages 367–375
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

Text difficulty evaluation traditionally used surface-level formulas like DCRS (Dale and Chall, 1948), FKGL (Kincaid et al., 1975), Dawood and El-Heeti (Al-Dawsari, 2004), AARI (Al Tamimi et al., 2014), and OSMAN (El-Haj and Rayson, 2016). The development of readability corpora enabled richer statistical and neural modeling approaches.

These corpus-driven advances have been most prominent in English, with resources including WeeBit (Vajjala and Meurers, 2012), Newsela (Xu et al., 2015), Cambridge (Xia et al., 2016), OneStopEnglish (Vajjala and Lučić, 2018) (document-level), S1131 (Štajner et al., 2017), CEFR-SP (Arase et al., 2022) (sentence-level), and cross-lingual corpora MDTE (De Clercq and Hoste, 2016), CompDS (Brunato et al., 2018). For Arabic, efforts such as (Hazim et al., 2022) have facilitated Arabic readability annotations, leading to the development of resources spanning multiple granularities: documents (Arability (Al-Khalifa and Al-Ajlan, 2010), DLI (Forsyth, 2014), Taha/Arabi21 (Taha-Thomure, 2017), ZAEBUC (Habash and Palfreyman, 2022), QAES (Bashendy et al., 2024)), sentences (README++ (Naous et al., 2024), DARES (El-Haj et al., 2024), BAREC (Elmadani et al., 2025b)), and words (KELLY (Kilgarriff et al., 2014), SAMER (Al-Khalil et al., 2020), Arabic Vocab Profile (Soliman and Familiar, 2024), extended SAMER (Alhafni et al., 2024)).

Building on these corpus development efforts, broader research has focused on Arabic readability modeling using diverse strategies (Liberato et al., 2024). To advance this field further and provide a standardized evaluation framework, the BAREC Shared Task 2025 (Elmadani et al., 2025a) introduces 19-level fine-grained readability prediction with three tracks: Strict (BAREC corpus only), Constrained (BAREC & SAMER corpora), and Open (any public data). We participate in the Strict Track for sentence-level assessment, predicting Arabic sentence difficulty on a 19-point scale (1 = easiest, 19 = hardest) using models trained exclusively on BAREC training data.

3 System Overview

We experiment with CAMeLBERTMix_MLP and CAMeLBERTWCE_MLP, which combine contextual embeddings from Arabic BERT models with a lightweight multilayer perceptron (MLP) classifier. CAMeLBERTMix_MLP employs bert-base-arabic-camelbert-mix [2] (Inoue et al., 2021), a general-purpose encoder trained on a mix of modern standard, dialectal, and classical Arabic, while CAMeLBERTWCE_MLP uses readability-camelbert-word-CE [3] (Elmadani et al., 2025b), fine-tuned on the same dataset as this shared task. Sentences are tokenized with a maximum length of 256 tokens, encoded, and aggregated via mean pooling over attention-masked hidden states to yield 768 dimensional embeddings. These embeddings are normalized using RobustScaler and fed into the MLP, which was selected via 5-fold stratified cross-validation. The best-performing configuration is a two-layer architecture (256-128 neurons), ReLU activation, L2 regularization ($\alpha = 0.01$), trained with Adam optimization (initial learning rate 10^{-4}, adaptive scheduling), batch size 64, and a maximum of 300 epochs.

P_CAMeLBERTWCE_MLP architecture implements a progressive multilayer perceptron (MLP) trained with a curriculum learning strategy. The model is trained sequentially through multiple stages of increasing granularity (3-5-7-19 levels), utilizing the same 768 dimensional embeddings as CAMeLBERTWCE_MLP. The 19-level ground truth labels are collapsed into intermediate targets using custom binning strategies to create more balanced distributions: 3-level bins [0, 7, 13, 19], 5-level bins [0, 4, 8, 12, 16, 19], and 7-level bins [0, 3, 6, 9, 12, 15, 17, 19]. Training begins with a simple 3-level classifier (2-layer architecture with 256 and 128 neurons), then progresses to 5-level and 7-level classifiers, and finishes with a 19-level classifier (3-layer architecture with 512, 256, and 128 neurons). The stage-specific architecture scales with task complexity. A weight transfer mechanism preserves learned hidden layer representations between stages. The model is optimized using Adam with an adaptive learning rate (initial $\eta = 10^{-3}$, reduced to 5×10^{-4} during transfer phases), a batch size of 64, and L2 regularization ($\alpha \in [0.008, 0.01]$) and a maximum of 100-300 epochs per stage depending on complexity.

PFT_CAMeLBERTWCE implements a Progressive CamelBERT (Inoue et al., 2021) Fine-tuning approach that fine-tunes the CAMeL-Lab/readability-camelbert-word-CE transformer model through curriculum learning stages [3, 5, 7, 19]. The ap-

[2] https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-mix

[3] https://huggingface.co/CAMeL-Lab/readability-camelbert-word-CE

proach learns dynamic label mappings from dataset annotations rather than using fixed binning strategies like we did in P_CAMeLBERTWCE_MLP. Training begins with 3-level classification using a dropout-regularized classification head (dropout=0.3), progressively transferring the fine-tuned BERT encoder weights to subsequent stages while initializing fresh classification heads for each target granularity. Training uses AdamW optimization with linear warmup scheduling, adaptive learning rates (2e-5 initial, 1e-5 for transfer stages). The intuition is that learning coarse readability distinctions (3-5-7 levels) first provides foundational representations that will improve fine-grained (19 levels) classification performance.

FT_CAMeLBERTWCE serves as an ablation study to test the core hypothesis behind PFT_CAMeLBERTWCE. It eliminates the progressive curriculum learning stages to perform direct, end-to-end fine-tuning of the CAMeL-Lab/readability-camelbert-word-CE model on the full 19-level classification task. The motivation for this simpler approach is twofold. First, it questions whether a powerful pre-trained transformer inherently possesses the latent linguistic understanding to discern fine-grained readability distinctions without being guided through coarser labels. Second, it tests if the considerable computational and architectural overhead of multi-stage progressive training is justified, or if a single-stage model can achieve comparable performance more efficiently. To ensure a fair comparison, FT_CAMeLBERTWCE retains the same optimization strategy (AdamW, linear warmup) and regularization (dropout=0.3) as the final stage of PFT_CAMeLBERTWCE. Thus allowing us to directly attribute any performance differences to the presence or absence of the curriculum learning framework, rather than other hyperparameters.

4 Experimental Setup:

4.1 Dataset

We used the Balanced Arabic Readability Evaluation Corpus (BAREC)[4], a large-scale, fine-grained corpus containing over 69,000 sentences from 1,922 documents. As illustrated in Table 1, the corpus spans 19 readability levels, from kindergarten (1) to postgraduate (19), which can also be collapsed into coarser 7, 5, or 3 readability levels. For more details on sentence readability annotation, refer to the BAREC readability annotation guidelines by (Habash et al., 2025).

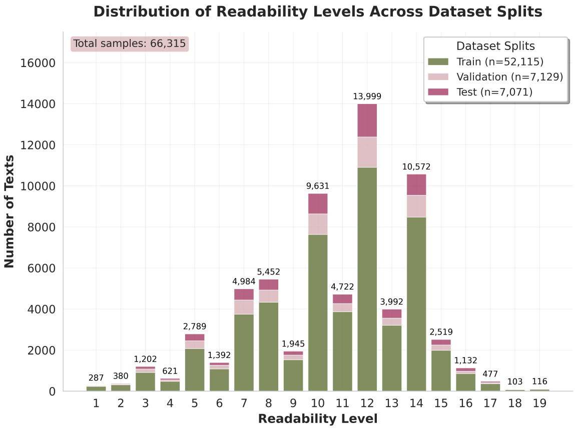


Figure 1: Distribution of the train/validation/test split in the BAREC corpus by level of readability

Level	Arabic	Translation	Reasoning
1	'نعم'	“Yes”	Simple single-word
19	'أو صورة مثَلت في النفس من أملي'	“Or an image represented in the soul from my hope.”	Complex poetic expression, rich in imagery

Table 1: Comparison of linguistic complexity between beginner and advanced Arabic expressions.

4.2 Preprocessing

Before training, we applied a multi-stage cleaning pipeline to ensure text consistency and quality. This involved removing sentences with missing values or placeholders (e.g., #NAME?), collapsing excessive whitespace, and dropping exact duplicates. For Arabic processing, we utilized CAMeL Tools (Obeid et al., 2020), a comprehensive suite of NLP resources for morphological analysis, disambiguation, dialect identification, normalization, and tokenization [5]. A key step was dediacritization to remove short vowels and phonetic marks witch are infrequent in modern Arabic and can introduce noise in NLP tasks. Dediacritization allowed us to focus on underlying lexical forms (e.g., كِتَابٌ جَمِيلٌ was converted to كتاب جميل, or “A beautiful book”). The pipeline removed 2.5-5% of rows, retaining 95-97% of the dataset across splits (52k train, 7.1k validation, 7k test). The final distribution of these readability levels is shown in Figure 1.

[4] BAREC corpus: https://huggingface.co/datasets/CAMeL-Lab/BAREC-Shared-Task-2025-sent

[5] CAMeL Tools:https://github.com/CAMeL-Lab/camel_tools

4.3 Metrics

The BAREC shared task [6] organizers define the readability prediction problem as an ordinal classification task and adopt the following evaluation [7] measures: **Quadratic Weighted Kappa (QWK) (Cohen, 1968):** The primary evaluation metric for the shared task. It's an extension of Cohen's Kappa that measures agreement between predicted and reference labels, applying a quadratic penalty to larger misclassifications. **Accuracy (Acc):** The percentage of exact matches between predicted and gold labels on the 19-level scale scale Acc^{19}. Variants Acc^{7}, Acc^{5}, and Acc^{3} are computed on collapsed 7-, 5-, and 3-level versions of the scale, respectively. **Adjacent Accuracy (±1 Acc^{19}):** counts predictions as correct if they are either exact matches or differ by at most one level from the true label. **Average Distance (Dist):** Also referred to as Mean Absolute Error (MAE), it captures the mean absolute difference between predicted and reference labels.

5 Results

Our team's runs (LIS in (Elmadani et al., 2025a)), were evaluated on Codabench [8]. Table 2 shows results on the blind test set, using the QWK and Acc^{19} metrics (Section 4.3). Initial experiments with MLP classifiers on embeddings demonstrated the importance of domain-specific pretraining. The general-purpose `CAMeLBERTMix_MLP` performed poorly, while `CAMeLBERTWCE_MLP` utilizing embeddings from a readability focused model demonstrated marked improvement, confirming the efficacy of task-specific pretraining. Introducing curriculum learning (`P_CAMeLBERTWCE_MLP`) provided only a marginal gain, indicating that while progressive binning can stabilize training, its impact is limited with a frozen encoder. In contrast, curriculum learning proved more beneficial in the fine-tuning setting. `PFT_CAMeLBERTWCE`, which progressively fine-tunes the encoder through increasingly granular label spaces, outperformed both MLP-based models. Finally, direct fine-tuning without curriculum (`FT_CAMeLBERTWCE`) achieved the best overall QWK (82.4) and second-best Acc^{19} (57.5), slightly surpassing the progressive strategy.

To better understand model behavior, we conducted an error analysis of the two fine-tuning approaches (`PFT_CAMeLBERTWCE` and `FT_CAMeLBERTWCE`) on the preliminary test set. We reported conditional error rates by readability level, calculated as the percentage of incorrect predictions within each readability level. Additionally, we provided a confusion matrix in Figures 3 and 4 (Appendix A) for both the best-performing model (`FT_CAMeLBERTWCE` and second-best model (`PFT_CAMeLBERTWCE`) to illustrate prediction tendencies and error patterns in greater detail. While `FT_CAMeLBERTWCE` achieved superior overall performance, the error rate analysis (Figure 2, Appendix A) reveals that `PFT_CAMeLBERTWCE` demonstrates lower error rates for specific readability levels (3-6, 9, 11, 12, 16, 17), suggesting that curriculum learning provides targeted improvements for certain readability levels despite lower aggregate performance. This level-specific analysis complemented the aggregate metrics by revealing where each model struggled most in distinguishing between readability levels, providing insights into the model's systematic biases and failure modes.

6 Conclusion

We evaluated neural approaches for Arabic sentence readability assessment, comparing MLP classifiers using CamelBERT embeddings with transformer fine-tuning methods. Task-specific pretraining proved to be essential, general embeddings failed while readability-focused ones improved performance substantially. Curriculum learning provided marginal gains with frozen encoders but helped stabilize fine-tuning. Direct CamelBERT fine-tuning (`FT_CAMeLBERTWCE`) achieved best results (QWK = 82.4, Acc = 57.5), surpassing baselines and slightly outperforming progressive fine-tuning. Our experiments highlight three key insights. First, task-specific pretraining is crucial for Arabic readability assessment, with domain-aligned representations significantly outperforming general-purpose embeddings. Second, curriculum learning offers modest but situational benefits. Third, direct fine-tuning remains both efficient and effective, achieving state-of-the-art performance without complex training strategies. Error analysis revealed systematic biases across difficulty levels, as illustrated in Figure 2 (Appendix A). Future work will explore hybrid architectures combining transform-

[6] BAREC shared task: `https://barec.camel-lab.com/sharedtask2025`

[7] Evaluation metrics: `https://github.com/CAMeL-Lab/barec_analyzer/tree/main`

[8] Codabench: `https://www.codabench.org/`

Model	Description	QWK	Acc[19]
Baseline	Competition baseline	81.5	**58.1**
CAMeLBERTMix_MLP	Word Embedding + MLP	41.2	21.2
CAMeLBERTWCE_MLP	Word Embedding + MLP	80.5	55.7
P_CAMeLBERTWCE_MLP	Word Embedding + Progressive training of MLP	80.7	55.9
PFT_CAMeLBERTWCE	Progressive Fine-tuning	<u>82.0</u>	56.7
FT_CAMeLBERTWCE	Standard Fine-tuning	**82.4**	<u>57.5</u>

Table 2: Model performance on the blind test measured using Quadratic Weighted Kappa (QWK) and Accuracy (Acc[19]). P = progressive training strategy; PFT = progressive fine-tuning; FT = standard fine-tuning; MLP = multi-layer perceptron. CAMeLBERTWEC = CAMeLBERT-Word-CE; CAMeLBERTMix = CAMeLBERT-Mix. **Bold** indicates the best result; <u>underlined</u> indicates the second-best.

ers with linguistic features and multi-agent frameworks.

Limitations

This study was limited to transformer-based approaches and word embeddings. Incorporating explicit linguistic features (such as syntactic complexity, lexical diversity, and discourse markers) could complement these neural representations and potentially improve both readability prediction accuracy and model explainability.

References

M Al-Dawsari. 2004. The assessment of readability books content (boys-girls) of the first grade of intermediate school according to readability standards. *Sultan Qaboos University, Muscat.*

Hend S Al-Khalifa and Amani A Al-Ajlan. 2010. Automatic readability measurements of the arabic text: An exploratory study. *Arabian Journal for Science and Engineering*, 35(2 C):103–124.

Muhamed Al-Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for standard arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062.

Abdel Karim Al Tamimi, Manar Jaradat, Nuha Al-Jarrah, and Sahar Ghanem. 2014. Aari: automatic arabic readability index. *Int. Arab J. Inf. Technol.*, 11(4):370–378.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The samer arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093.

Yuki Arase, Satoru Uchida, and Tomoyuki Kajiwara. 2022. Cefr-based sentence difficulty annotation and assessment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6206–6219.

Ion Madrazo Azpiazu and Maria Soledad Pera. 2019. Multiattentive recurrent neural network architecture for multilingual readability assessment. *Transactions of the Association for Computational Linguistics*, 7:421–436.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Eman Zahran, Hamdo Elhuseyin, Tamer Elsayed, Walid Massoud, and Houda Bouamor. 2024. Qaes: First publicly-available trait-specific annotations for automated scoring of arabic essays. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 337–351.

Dominique Brunato, Lorenzo De Mattei, Felice Dell'Orletta, Benedetta Iavarone, Giulia Venturi, and 1 others. 2018. Is this sentence difficult? do you agree? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 2690–2699. Association for Computational Linguistics.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Kevyn Collins-Thompson and Jamie Callan. 2004. Information retrieval for language tutoring: An overview of the reap project. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 544–545.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Orphée De Clercq and Véronique Hoste. 2016. All mixed up? finding the optimal feature set for general readability prediction and its application to english and dutch. *Computational Linguistics*, 42(3):457–490.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. *arXiv preprint arXiv:2006.00377.*

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.

Anya Amel Nait Djoudi, Patrice Bellot, and Adrian-Gabriel Chifu. 2025. Bioreadnet: A transformer-driven hybrid model for target audience-aware biomedical text readability assessment. In *Proceedings of the 2025 ACM Symposium on Document Engineering*, DocEng '25, page 1–10. ACM.

Mahmoud El-Haj and Paul Rayson. 2016. Osman a novel arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255.

Mo El-Haj, Sultan Almujaiwel, Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2024. Dares: Dataset for arabic readability estimation of school materials. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context@ LREC-COLING 2024*, pages 103–113.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Jonathan Neil Forsyth. 2014. *Automatic readability prediction for modern standard Arabic*. Ph.D. thesis, Brigham Young University. Department of Linguistics and English Language.

Susannah Fox and Maeve Duggan. 2013. Health online 2013. *Health*, 2013:1–55.

Nizar Habash and David Palfreyman. 2022. Zaebuc: An annotated arabic-english bilingual writer corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 79–88.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. *arXiv preprint arXiv:2210.10672*.

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.

Adam Kilgarriff, Frieda Charalabopoulou, Maria Gavrilidou, Janne Bondi Johannessen, Saussan Khalil, Sofie Johansson Kokkinakis, Robert Lew, Serge Sharoff, Ravikiran Vadlapudi, and Elena Volodina. 2014. Corpus-based vocabulary lists for language learners for nine languages. *Language resources and evaluation*, 48(1):121–163.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Juan Piñeros Liberato, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2024. Strategies for arabic readability modeling. *arXiv preprint arXiv:2407.03032*.

Tarek Naous, Michael J Ryan, Anton Lavrouk, Mohit Chandra, and Wei Xu. 2024. Readme++: Benchmarking multilingual language models for multi-domain readability assessment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2024, page 12230.

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.

Xinying Qiu, Yuan Chen, Hanwu Chen, Jian-Yun Nie, Yuming Shen, and Dawei Lu. 2021. Learning syntactic dense embedding with correlation graph for automatic readability assessment. 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 3013–3025, Online. Association for Computational Linguistics.

Rasha Soliman and Laila Familiar. 2024. Creating a cefr arabic vocabulary profile: A frequency-based multi-dialectal approach. *Critical Multilingualism Studies*, 11(1):266–286.

Sanja Štajner, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2017. Automatic assessment of absolute sentence complexity. In *Proceedings of the 26th international joint conference on artificial intelligence, ijcai*, volume 17, pages 4096–4102.

Hanada Taha-Thomure. 2017. Arabic language text leveling(). *Educational Book House ()*.

Sowmya Vajjala and Ivana Lučić. 2018. Onestopenglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 297–304.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the seventh workshop on building educational applications using NLP*, pages 163–173.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

A Model performance analysis

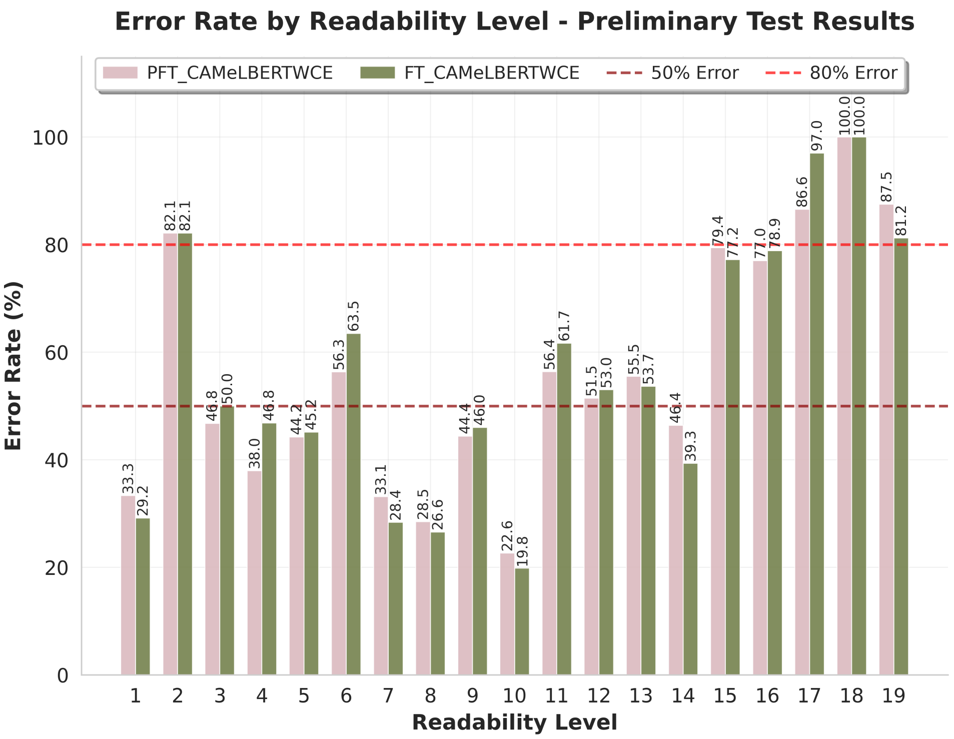


Figure 2: Error Rate by readability level on the preliminary test set

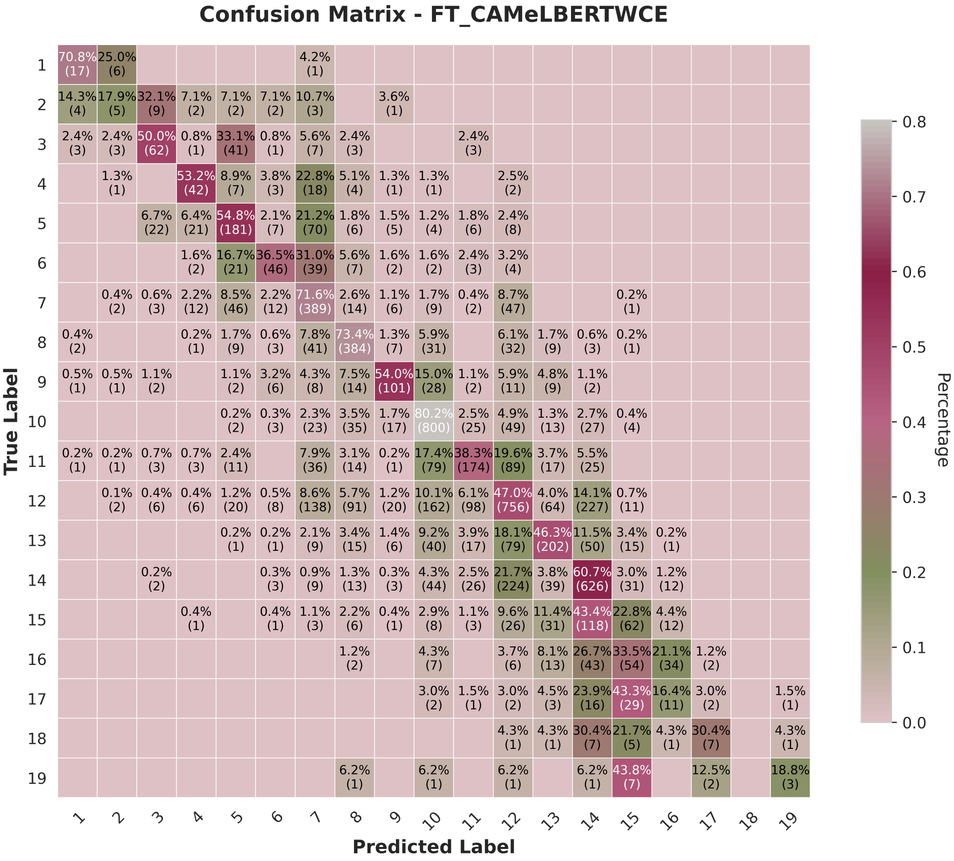


Figure 3: Prediction of FT_CAMeLBERTWCE on the preliminary test set

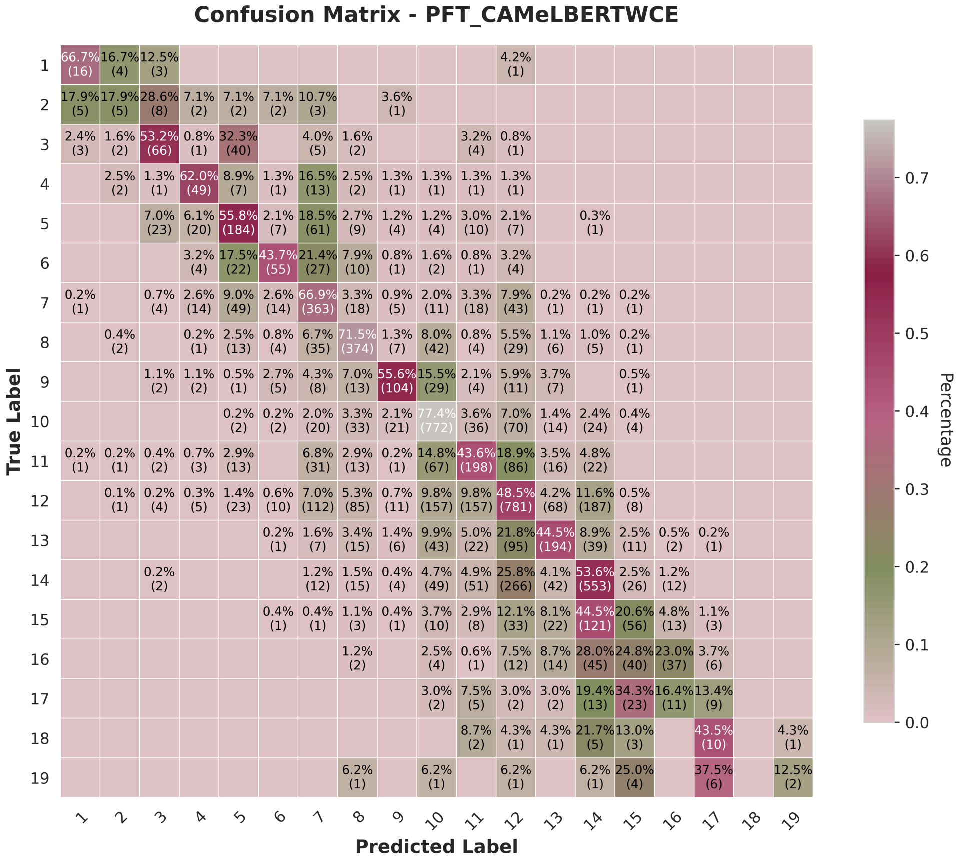


Figure 4: Prediction of PFT_CAMeLBERTWCE on the preliminary test set

ImageEval 2025: The First Arabic Image Captioning Shared Task

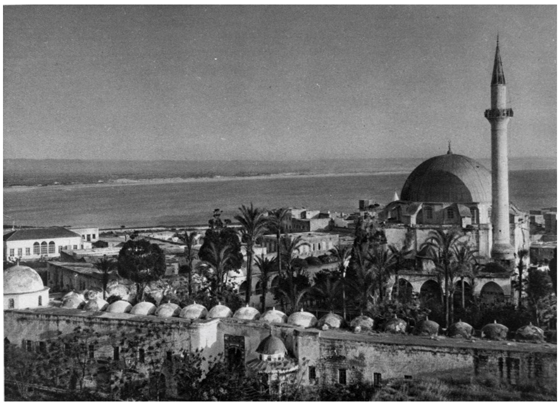
Ahlam Bashiti[1], **Alaa Aljabari**[1], **Hadi Hamoud**[3], **Md. Rafiul Biswas**[2] , **Bilal Shalash**[3], **Mustafa Jarrar**[2,1], **Fadi Zaraket**[3,4], **George Mikros**[2], **Ehsaneddin Asgari**[2], **Wajdi Zaghouani**[5]

[1]Birzeit University, [2]Hamad Bin Khalifa University, [3]American University of Beirut,

[4]Arab Center for Research and Policy Studies, [5]Northwestern University in Qatar

Abstract

We present ImageEval 2025, the first shared task dedicated to Arabic image captioning. The task addresses the critical gap in multimodal Arabic NLP by focusing on two complementary subtasks: (1) creating the first open-source, manually-captioned Arabic image dataset through a collaborative datathon, and (2) developing and evaluating Arabic image captioning models. A total of 44 teams registered, of which eight submitted during the test phase, producing 111 valid submissions. Evaluation was conducted using automatic metrics, LLM-based judgment, and human assessment. In Subtask 1, the best-performing system achieved a cosine similarity of 65.5, while in Subtask 2, the top score was 60.0. Although these results show encouraging progress, they also confirm that Arabic image captioning remains a challenging task, particularly due to cultural grounding requirements, morphological richness, and dialectal variation. All datasets, baseline models, and evaluation tools are released publicly to support future research in Arabic multimodal NLP.



Manual Caption (Culturally Relevant):
صورة تظهر مسجد الجزّار في مدينة عكا الساحلية في فلسطين، أحد أبرز المعالم العثمانية بقبته ومئذنته البارزتين، تحيط به بيوت وأسوار قديمة، ما يعكس الطابع الحضاري والتاريخي للمدينة.
Translation: Al-Jazzar Mosque in Acre, Palestine, a major Ottoman landmark with its dome and minaret, surrounded by old houses and city walls reflecting the city's history.

Generated Caption (Culturally Irrelevant):
صورة تظهر منظرًا معماريًا قديمًا لمدينة ساحلية، تتضمن مسجدًا كبيرًا بقبته ومئذنته، محاطاً بأشجار النخيل ومباني منخفضة، مع البحر في الخلفية.
Translation: A coastal city view with a mosque, palm trees, and low-rise buildings by the sea.

Figure 1: Comparison of captions for the same image. The manual caption is culturally relevant, while the generated caption lacks cultural specificity.

1 Introduction

Image captioning, the automatic generation of natural language descriptions for visual content (Hossain et al., 2019), represents a fundamental challenge at the intersection of computer vision and natural language processing (Saraswat et al., 2024). While significant progress has been achieved for high-resource languages, particularly English, Arabic image captioning remains severely underexplored despite Arabic being spoken by over 400 million people worldwide (Mohamed et al., 2023b).

The challenges of Arabic image captioning extend beyond typical technical hurdles. Arabic's rich morphology, diverse dialectal variations, short vowel omissions, right-to-left script, and cultural diversity require specialized approaches that consider linguistic, cultural, and contextual factors (Jarrar et al., 2023b). Moreover, the lack of large-scale, high-quality Arabic image-caption datasets has hindered progress in this domain.

To address these challenges and highlight the unique issues in Arabic image captioning, we organized the ImageEval 2025 shared task, which comprised two complementary subtasks: Subtask

1, a collaborative image captioning datathon, and Subtask 2, an evaluation of Arabic image captioning models. The task design follows principles of cultural and linguistic authenticity, methodological diversity, and rigorous evaluation. Subtask 1 ensured that captions accurately reflected the perspectives and contextual norms of Arabic speakers, moving beyond direct translations from other languages. Figure 1 illustrates a comparison between manual and generated captions for the same image, where the manual caption reflects cultural context and historically specific information, whereas the generated caption (by GPT-5 mini) provides a general description with limited cultural relevance.

Subtask 2 encouraged participating teams to experiment with a broad range of modeling strategies, including zero-shot, few-shot, and fully supervised approaches. Model outputs were evaluated using a combination of widely adopted automatic metrics for image captioning, such as BLEU (Papineni et al., 2002) and cosine similarity (Sharif et al., 2020), as well as LLM-based assessments that capture semantic correctness and contextual appropriateness (Zhang et al., 2025). In addition, human evaluation was conducted to provide a complementary benchmark, focusing on fluency, cultural adequacy, and alignment with the visual content, thereby assessing subjective quality aspects not captured by automatic metrics. The shared task explicitly addresses challenges such as dataset scarcity, morphological complexity, cultural specificity, metric suitability, and resource constraints in Arabic NLP research.

This paper presents a comprehensive overview of ImageEval 2025, including our motivation, task design principles, data collection methodology, evaluation framework, baseline models, and analysis of participant approaches and results. Our contributions include:

- Introduce the first large-scale shared task for Arabic image captioning, combining collaborative data creation with competitive model evaluation.
- Comprehensive evaluation framework incorporating automatic metrics, LLM-based assessment, and human evaluation.
- Analysis of cultural and linguistic challenges specific to Arabic image captioning.
- Release all resources, including datasets, evaluation tools, and baseline models, as open-source.

The rest of the paper is organized as follows: Section 2 reviews related work, Section 3 presents the shared task overview, Section 4 describes the evaluation methodology, Sections 5 and 6 detail Subtasks 1 and 2, Section 7 discusses challenges and insights, Section 8 covers impact and future directions, and Section 9 concludes the paper.

2 Related Work

2.1 Evolution of Image Captioning

Early image captioning relied on template-based (Farhadi et al., 2010; Kulkarni et al., 2013) and retrieval methods (Devlin et al., 2015; Ordonez et al., 2011), but the field was revolutionized by the adoption of encoder-decoder frameworks, where CNNs extract image features and RNNs or LSTMs generate captions (Stefanini et al., 2023; Ming et al., 2022; Hossain et al., 2018; Verma et al., 2023). The introduction of attention mechanisms allowed models to focus on salient image regions, improving caption relevance and fluency (Yu et al., 2019; Liu et al., 2020; Yan et al., 2021; Wang et al., 2020; Gao et al., 2020). Transformer-based models further advanced the field by enabling parallel processing and capturing long-range dependencies, leading to state-of-the-art results on benchmarks like MSCOCO (Yu et al., 2019; Yan et al., 2021; Xian et al., 2022; Parvin et al., 2023). Recent vision-language models such as CLIP, BLIP, and GPT-4V leverage large-scale pretraining and multimodal fusion, achieving remarkable performance and enabling new applications in accessibility and content retrieval (Khodave and Powar, 2025; Cho and Oh, 2023; Betala and Chokshi, 2024; Nguyen et al., 2023).

2.2 Multilingual and Cross-lingual Image Captioning

Multilingual image captioning has gained traction, with datasets like COCO-CN and Crossmodal-3600 supporting multiple languages (Cho and Oh, 2023; Li et al., 2019; Song et al., 2023). Most research, however, still focuses on resource-rich languages, with English dominating available data and benchmarks (Khodave and Powar, 2025; Cho and Oh, 2023; Li et al., 2019; Song et al., 2023). Cross-lingual transfer approaches, such as using visual pivots or synthetic data, have shown promise in generating captions for low-resource languages

(Al-Buraihy and Wang, 2024; Zhang et al., 2023; Hitschler et al., 2016; Song et al., 2023). Recent models employ transformer architectures and reinforcement learning to improve semantic and stylistic alignment across languages (Al-Buraihy and Wang, 2024; Zhang et al., 2023; Song et al., 2023). Despite these advances, morphologically complex languages like Arabic remain underexplored, and open-source models often lag behind proprietary systems in multilingual performance (Cho and Oh, 2023; Zha et al., 2022; Betala and Chokshi, 2024; Song et al., 2023).

2.3 Arabic NLP and Multimodal Processing

Arabic NLP presents unique challenges due to its rich morphology, complex script, and wide dialectal variation (Nayouf et al., 2023). Moreover, the meaning of many Arabic words can shift significantly depending on context (Jarrar, 2021; Akra et al., 2025). Several studies have been conducted on Arabic image captioning (Elbedwehy and Medhat, 2023; Emami et al., 2022b; ElJundi et al., 2020; Afyouni et al., 2021; Alsayed et al., 2023; Hejazi and Shaalan, 2021). While significant progress has been made in text-only Arabic NLP, multimodal applications, especially image captioning, are still nascent. Recent studies have proposed transformer-based and hybrid models for Arabic image captioning, often leveraging pre-trained language models such as AraBERT, MARBERT, and CamelBERT (Badarneh et al., 2025; Yu et al., 2019; Elbedwehy and Medhat, 2023; Emami et al., 2022b; Afyouni et al., 2021; Alsayed et al., 2023; Sabri, 2021). These models have demonstrated improved performance over translation-based approaches, but the lack of large, high-quality Arabic datasets remains a major bottleneck (Elbedwehy and Medhat, 2023; Emami et al., 2022b; ElJundi et al., 2020; Afyouni et al., 2021; Alsayed et al., 2023; Hejazi and Shaalan, 2021). Comparative studies highlight the importance of tailored preprocessing and feature extraction for Arabic, with some models achieving BLEU-4 scores up to 0.16, outperforming earlier work (Elbedwehy and Medhat, 2023; Alsayed et al., 2023; Sabri, 2021; Hejazi and Shaalan, 2021).

2.4 Shared Tasks in Multimodal NLP

Shared tasks and benchmarks such as MSCOCO, VQA, and COCO-CN have been instrumental in advancing image captioning and multimodal NLP (Khodave and Powar, 2025; Cho and Oh, 2023; Li et al., 2019; Betala and Chokshi, 2024). These challenges foster innovation, provide standardized evaluation, and drive the development of robust models (Khodave and Powar, 2025; Cho and Oh, 2023; Li et al., 2019; Betala and Chokshi, 2024). In addition, several Arabic shared tasks have addressed a range of NLP tasks, including named entity recognition (Jarrar et al., 2024, 2023a), language understanding (Khalilia et al., 2024), and dialect identification (Abdul-Mageed et al., 2024, 2023), demonstrating the value of community-driven evaluation across diverse language technologies. However, no major shared task has specifically targeted Arabic image captioning, highlighting a significant gap and an opportunity for future community-driven efforts (Cho and Oh, 2023; Betala and Chokshi, 2024; Sabri, 2021).

3 Shared-task Overview

The ImageEval 2025 shared task comprises two primary subtasks: Subtask 1, the Image Captioning Datathon, and Subtask 2, the Image Captioning Models Evaluation. Subtask 1 focuses on the manual creation of Arabic image captions, requiring participants to produce natural, culturally appropriate, and contextually aligned descriptions. Captions must be written manually, without the use of generative AI tools, and participants were provided with minimal contextual information about the images. This guidance helps teams generate meaningful captions that accurately reflect the content and cultural context of each image.

Subtask 2 evaluates the performance of Arabic image captioning models. Participants are allowed to use external datasets and retrieval-augmented generation (RAG) approaches; however, the submitted system must rely entirely on the provided dataset for evaluation. This requirement ensures a standardized and fair comparison across participating models.

The shared task received 44 registrations, and during the test phase, 8 teams submitted a total of 111 entries (109 submissions for Subtask 1 and 2 submissions for Subtask 1). In addition, 8 system description papers were submitted and all were accepted. To facilitate consistent evaluation and scoring of submissions, we employed Codabench[12], a well-established platform for shared-task evaluation. Furthermore, we established and shared a dedicated web page for the shared task, providing

[1] https://www.codabench.org/competitions/9447/

[2] https://www.codabench.org/competitions/9450/

participants with guidelines and detailed information as a reference [3]. Table 1 presents a detailed overview of the participating teams, listed in alphabetical order, along with their affiliations and the subtasks in which they participated.

4 Evaluation

4.1 Human Evaluation

We selected approximately 5% of the test data and applied four qualitative metrics to all participating teams. Each metric was rated on a scale from 1 (lowest) to 4 (highest):

- **Cultural Relevance** – Measures whether the description reflects cultural specificity and provides contextual information related to the scene.
- **Conciseness** – Assesses whether the description conveys information directly and succinctly, without unnecessary repetition or dispersion of details.
- **Completeness** – Evaluates the extent to which the description covers all aspects of the image, including events, entities, and relevant elements.
- **Accuracy** – Measures whether the description contains correct information, free from factual or conceptual errors.

4.2 Automatic Metrics

The task considered the following metrics for automatic evaluation of submissions.

- **BLEU** measures n-gram ($n \in [1,4]$) overlap between generated and reference captions, and applies smoothing for sparse higher-order n-grams.
- **ROUGE scores**: (ROUGE-1, ROUGE-2, and ROUGE-L) are recall-oriented; they measure how many reference n-grams are recovered by the candidate caption and the longest common subsequence.
- **Cosine similarity**: compares the angular distance between vector representations of the captions. For this task, we used term-frequency–inverse-document-frequency (TF–IDF) vectors, where terms are n-grams ($n \in [1,4]$) and each caption is a document.
- **Jaccard Similarity**: calculates the intersection over union of unique word sets, providing a set-based overlap measure.
- **Lin Similarity**: is an information-theoretic metric that computes twice the ratio of the information content (IC) of the least common subsumer of both captions, divided by the sum of the IC of both captions.

4.3 LLM as a Judge

We incorporated LLM as a judge in the scoring pipeline. Specifically, we employed the OpenAI GPT-4o model through its API, with a fixed random seed of 42, and an inference temperature of 0.0 to ensure reproducibility, using a task-specific system prompt (Appendix B).

For each (candidate, reference) caption pair, we provided a structured prompt and instructed the LLM to assign an integer score between 1 and 10, where 1 is lowest and 10 is highest similarity. The evaluation criteria emphasized semantic accuracy, relevance, and fluency of the candidate caption compared to the reference one.

Model outputs were parsed to reduce ambiguity, and evaluations were executed concurrently for efficiency. Final submission scores were obtained by averaging across all pairs and mapping results to a normalized $[0, 100]$ scale.

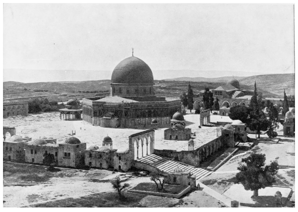
5 Subtask 1: Image Captioning Datathon

Images depict diverse visual scenes that require contextually rich and culturally informed descriptions, which motivated the Image Captioning Datathon (Subtask 1). This subtask aims to generate captions that are both linguistically natural and culturally appropriate for Arabic. Given an image I, the goal is to produce a caption C that accurately describes the content of I while reflecting Arabic language norms and cultural context. Participants were provided with a set of images and tasked with manually creating descriptive captions that emphasize meaning, context-awareness, and cultural grounding. Submissions were required in a CSV format, containing the corresponding image ID and the generated caption for each image in the test set. Figure 2 illustrates an example image along with its manually annotated caption.

[3] https://sina.birzeit.edu/image_eval2025/

Team	Affiliation	Subtask 1	Subtask 2
AZLU (Yassine et al., 2025)	Lebanese Univ., Birzeit Univ., Al Azhar Univ.	✓	
BZU-AUM (Alkhanafseh et al., 2025)	Birzeit Univ.	✓	
Averroes (Saeed et al., 2025)	Applied Innovation Center, Georgia Tech		✓
Phantom Troupe (Abu Horaira et al., 2025)	Chittagong Univ. of Engineering and Technology		✓
VLCAP (Elchafei and Fashwan, 2025)	Ulm Univ., Alexandria Univ.		✓
Codezone Research Group (Bichi et al., 2025)	Baba Ahmed Univ. Kano		✓
ImpactAi (Al-Qasem and Hendi, 2025)	Ggateway		✓
NU_Internship (Gaber et al., 2025)	Nile Univ., Ain Shams Univ., Alex. Univ.		✓

Table 1: Participating teams in ImageEval 2025 and their subtasks.



Arabic: صورة لساحة مسجد قبة الصخرة في الحرم الشريف

English Translation: An image of the courtyard of the Dome of the Rock Mosque in Al-Haram Al-Sharif.

Figure 2: Example image with corresponding caption.

5.1 Dataset

The dataset comprises 4,000 open-source images collected from multiple domains with careful consideration to ensure cultural relevance and to avoid sensitive or inappropriate content. The dataset was systematically partitioned into 16 batches, each containing 250 images.

The images represent a broad spectrum of Palestinian cultural and social contexts. They encompass everyday life, the activities of liberation movements including military training, the lived experiences of refugees, and significant historical and touristic landmarks. The selection process prioritized diversity of perspectives to produce a dataset that is both rich and representative.

For evaluation, two batches (500 images) with pre-existing manual annotations were specified as mandatory. These annotations served as the reference ground truth for assessing the quality of the captions generated by participating teams. All teams were required to submit captions for these batches. In addition, teams were allowed to select further batches for annotation, provided that any chosen batch was captioned in its entirety.

5.2 Annotation Guidelines

Annotation guidelines were developed to ensure consistency across participants. Alongside these guidelines, an annotation file was provided containing 250 images organized into five sheets of 50 images each. Each sheet included a short contextual description, a thumbnail preview, and a URL to the original high-resolution image.

Participants were instructed to write captions in Modern Standard Arabic (MSA), avoiding colloquial or dialectal forms. Each caption was required to be between 15 and 100 words (ideally around 50 words, written in 3 – 4 sentences). Captions were expected to be narrative in style, reflecting emotions, events, historical context, and cultural significance, rather than simply listing visible objects. To ensure quality and consistency, participants were required to perform all annotations manually without AI assistance and to develop their own detailed captioning guidelines for internal use.

5.3 Evaluation and Result

For Subtask 1, captions were evaluated using human evaluation (4.1), automatic metrics (4.2), and LLM as a judge (4.3). Since cosine similarity and LLM-based scores showed higher alignment with human evaluation, they were used for final ranking. The combined results are summarized in Table 2.

According to automatic metrics, BZU-AUM (Alkhanafseh et al., 2025) achieved the highest cosine similarity (65.53), while AZLU (Yassine et al., 2025) obtained the highest LLM Judge Score (41.53). Human evaluation results indicate BZU-AUM scored highest in cultural relevance (3.24) and completeness (3.08), whereas AZLU scored highest in conciseness (3.44) and accuracy (3.16).

The results indicate different annotation tendencies between the two teams, with BZU-AUM producing more complete and culturally relevant descriptions, while AZLU provided captions that were comparatively more concise and accurate.

5.4 Discussion

The teams approached manual captioning through varied annotation strategies, team composition, quality control practices, and cultural adaptation methods.

Teams	Automatic Evaluation				Human Evaluation			
	Rank (Cosine)	Cosine Similarity	Rank (LLM)	LLM Judge Score	Cultural Relevance	Conciseness	Completeness	Accuracy
AZLU	2	59.15	1	**41.53**	3.20	**3.44**	2.88	**3.16**
BZU-AUM	1	**65.53**	2	32.42	**3.24**	2.76	**3.08**	2.92

Table 2: Subtask 1 Results: Automatic Evaluation (Cosine Similarity, LLM Judge Score) and Human Evaluation (Cultural Relevance, Conciseness, Completeness, Accuracy).

Annotation strategies varied with emphasis on narrative richness and contextual detail, while the other team focused on brevity and precision. These tendencies are reflected in the completeness and cultural depth favoring one team versus prioritized conciseness and accuracy.

Team composition played a role in shaping annotation styles, as teams included native Arabic speakers with dialectal backgrounds and subject matter experts for historically or culturally sensitive images. Quality assurance reviewers were also engaged to enhance consistency.

Quality control measures centered on internal review processes to ensure that captions adhered to guidelines and maintained fluency. While *inter-annotator agreement was not systematically enforced across all teams*, they adopted informal checks for coherence and style alignment.

Cultural adaptation approaches were particularly important, as annotators sought to embed historical references, social practices, and cultural nuances in the captions. This emphasis helped maintain cultural relevance while ensuring captions extended beyond object description into meaningful narrative.

6 Subtask 2: Image Captioning Model Evaluation

Subtask 2 addresses the development of models for automatic Arabic image captioning. Given an image $i \in I$, the goal is to generate an Arabic caption c_i that is both *contextually accurate* and *culturally relevant*. Participants were provided with a curated dataset of manually annotated Arabic images, divided into training and test subsets. The training subset was shared for model development, while the test set was released later for caption generation. Submissions consisted of automatically generated captions C for each test image $i \in I$, and were evaluated against the ground truth captions using established automatic metrics (see Section 4.2) through Codabench.

6.1 Dataset

We prepared a curated dataset of $3,471$ manually annotated Arabic image-caption pairs, comprising $2,718$ images for training with ground-truth captions and 753 images reserved for final evaluation. The images capture a wide range of Palestinian cultural and social contexts, including everyday life, the activities of liberation movements such as military training, the experiences of refugees, and notable historical and touristic landmarks. The dataset is publicly available through Hugging Face[4][5].

6.2 Baselines

To establish performance benchmarks, we established two baselines on our human-annotated dataset: zero-shot and fine-tuning. The code for these baselines is publicly available on GitHub[6].

Zero-Shot Baseline

For the zero-shot baseline, we employed `Qwen2.5-VL-7B-Instruct` (Bai et al., 2025), a vision–language model with a unified image encoder and autoregressive text decoder. The model was applied directly in inference mode, without any task-specific fine-tuning, to assess its ability to generate Arabic captions "out-of-the-box."

A multimodal prompt was designed to combine (i) the raw image and (ii) an instruction in Arabic guiding the model to generate culturally appropriate captions (15–50 words). The exact prompt template is provided in Appendix A.1. Inference was performed with a maximum generation length of 128 tokens. Outputs were collected in a structured format to facilitate evaluation. The final results are summarized in Table 3

Fine-Tuned Baseline

The same model was fine-tuned on the dataset using supervised fine-tuning (SFT) with LLAMA-FACTORY. Each training instance was formatted

[4] Train: `SinaLab/ImageEval2025Task2TrainDataset`

[5] Test: `SinaLab/ImageEval2025Task2TestDataset`

[6] Baselines: GitHub Repository

as a two-turn conversation in which the human prompt contained the image and a request for description in Arabic, and the assistant response was the corresponding gold caption.

Fine-tuning was carried out with parameter-efficient adaptation using LoRA. We trained for 15 epochs with a batch size of 16 and a maximum sequence length of 1024 tokens. Optimization employed AdamW with a learning rate of 2×10^{-5} and a cosine decay schedule with warmup. Dropout was set to 0.1, and the LoRA configuration used a rank of 8 with scaling parameter $\alpha = 16$. Training and validation losses were monitored throughout, and checkpoints were saved regularly. The fine-tuned model is publicly available[7].

Fine-tuning consistently improved performance across all evaluation metrics compared to the zero-shot baseline, as shown in Table 3, confirming the effectiveness of task-specific adaptation for Arabic image captioning.

Baseline	BLEU-1	BLEU-4	Cosine Similarity	LLM Judge (%)
Zero-shot	0.0992	0.0133	0.5577	27.11
Fine-tuned	0.1698	0.0305	0.5846	30.82

Table 3: Baseline performance on the dataset.

6.3 Participant Systems

All teams adapted pretrained vision–language models. They relied on translation, fine-tuning, or augmentation, and differed in how they restructured the captioning pipeline.

Averroes (Saeed et al., 2025) employs a two-stage pipeline, where one Qwen2.5-VL-7B based model generates detailed descriptions and another refines captions. They augmented training data with AyaVision8B and used BLEU scores to validate and pair with randomized image transformations. Its key contribution is systematic augmentation that enhances diversity without distorting the data distribution.

Codezone Research Group (Bichi et al., 2025) uses a zero-shot translation pipeline: BLIP generates English captions, which are translated to Arabic with M2M100. To ensure consistency in evaluation, the output is normalized by removing diacritics, Tatweel, and punctuation. Unlike others, it avoids fine-tuning, showcasing the viability of off-the-shelf models combined with robust translation.

ImpactAi (Al-Qasem and Hendi, 2025) proposed a region-aware captioning method based on the Region Features Transformer (CRAFT). The approach extracts a set of salient regions from each image using Faster R-CNN and encodes these region features through a transformer encoder–decoder architecture, paired with ArabGloss-BERT tokenization. This integration distinguishes it from other submitted methods.

Phantom Troupe (Abu Horaira et al., 2025) uses a translation-centered pipeline: Arabic captions are translated to English with the Qwen3-14B model for training and then back-translated at inference. They fine-tune Qwen2.5-VL-7B with LoRA for efficient adaptation. Its distinctive feature is the preservation of cultural nuances during translation while leveraging strong English captioning models.

NU_Internship (Gaber et al., 2025) adapted a vector store-based approach to enhance domain adaptability. They used Gemini-2.5 Flash, expanded the training data, and experimented with both zero-shot and fine-tuning, with and without RAG. To fuse the outputs of the top-performing models, they applied a meta-learning stacked ensemble using an LLM, selection guided by BLEU and cosine similarity metrics.

VLCAP (Elchafei and Fashwan, 2025) VLCAP is an Arabic image captioning framework that conditions generation on interpretable visual labels. A hybrid vocabulary is derived by extracting noun-like keywords from training dataset captions and augmenting them with over 21K translated Visual Genome concepts. Three retrieval experiments are conducted using mCLIP, AraCLIP, and Jina V4, where the top-k most relevant labels for each image are identified to construct the Arabic prompt for captions generation. These prompts, together with the original image, are provided to Qwen-VL and Gemini Pro Vision in separate settings. The best results are achieved when combining mCLIP for label retrieval with Gemini Pro Vision for caption generation, producing culturally coherent and contextually accurate Arabic captions, while AraCLIP with Qwen-VL excels in human-judged quality.

6.4 Evaluation and Results

Subtask 2 was assessed using the same three perspectives introduced earlier: automatic metrics (4.2), LLM as a judge (4.3), and human evaluation (4.1). Table 4 presents the comparative results for all participating teams.

VLCAP scored the highest cosine similarity

[7] Finetuned Baseline: Hugging Face

Teams	Automatic Evaluation				Human Evaluation			
	Rank (Cosine)	Cosine Similarity	Rank (LLM)	LLM Judge Score	Cultural Relevance	Conciseness	Completeness	Accuracy
Averroes	2	58.55	1	**33.97**	**3.63**	**3.43**	2.60	2.80
Codezone Research Group	6	38.30	6	15.14	1.10	2.03	1.47	2.03
ImpactAi	4	56.22	4	26.55	3.13	2.73	1.77	1.97
NU_Internship	5	55.32	5	24.87	2.57	2.97	2.13	2.23
Phantom Troupe	3	57.48	3	31.43	3.40	3.27	2.33	2.40
VLCAP	1	**60.01**	2	33.05	2.57	3.17	**2.67**	**2.97**

Table 4: Subtask 2 Results: Automatic Evaluation (Cosine Similarity, LLM Judge Score) and Human Evaluation (Cultural Relevance, Conciseness, Completeness, Accuracy).

(60.01), whereas Averroes ranked top with the LLM-Judge (33.97). Human evaluation highlights further distinctions: Averroes led in cultural relevance (3.63) and conciseness (3.43), while VLCAP ranked highest in completeness (2.67) and accuracy (2.97). Phantom Troupe also performed strongly, particularly in cultural relevance and conciseness.

6.5 Discussion

Submissions varied across model architectures, training, and fine-tuning strategies.

Model architectures were largely based on fine-tuning pretrained multilingual vision–language models, often with LoRA adapters for efficiency. Several teams relied on cross-lingual transfer by generating English captions and translating to Arabic, while one system introduced region-aware modeling with custom transformer components.

Training strategies included data augmentation, where image transformations and caption validation expanded the training set, and multi-stage pipelines that first produced detailed image descriptions before refining them into captions.

Arabic-specific optimizations focused on cultural and linguistic nuances during translation, dedicated Arabic tokenizers, and normalization to improve consistency in evaluation.

7 Challenges and Insights

Arabic image captioning faces significant challenges that stem from linguistic, cultural, and resource-related gaps. Unlike English, where large-scale datasets and robust models exist, Arabic research suffers from a severe shortage of high-quality, publicly available datasets (Emami et al., 2022a; Attai and Elnagar, 2020; Mohamed et al., 2023a; Kadaoui et al., 2025). Most available resources are translations from English rather than native Arabic captions, which fail to capture authentic linguistic patterns and cultural nuances (Ibrahim et al., 2025). This scarcity not only limits standardized benchmarking but also fragments research efforts, as scholars are forced to build small-scale datasets in isolation. Beyond resource limitations, Arabic itself introduces unique challenges due to its morphological richness, right-to-left script, connected character system (where OCR is needed), and extensive dialectal variation. These features make direct transfer of English-based methods ineffective, while translation-based approaches accumulate errors and degrade caption quality (Attai and Elnagar, 2020; Mohamed et al., 2023a). Cultural representation further complicates the task, as most image datasets are Western-centric and fail to reflect Arab cultural contexts, leading to mismatches between images and captions (Attai and Elnagar, 2020; Al-Buraihy et al., 2025). Addressing these challenges requires not only technical advances in preprocessing and modeling but also the creation of culturally authentic datasets tailored to Arabic's linguistic and social complexity.

ImageEval 2025 contributes to the benchmarking and further development of Arabic image captioning, offering a common ground for system comparison and incremental progress. By releasing the datasets, baseline models, and evaluation tools, this shared task aims to support the community and facilitate future research in Arabic multimodal NLP.

8 Future Directions

Building on the success and insights from ImageEval 2025, we identify several promising directions for future research and development in Arabic image captioning.

Future research should prioritize the development of evaluation metrics that more effectively capture Arabic morphological complexity, cultural nuances, and semantic variability, addressing limitations of current automatic measures. Addressing

Arabic dialectal diversity is another critical area, requiring models capable of adapting to regional linguistic variations and code-switching phenomena. Furthermore, enhancing cross-lingual transfer learning from high-resource languages while maintaining Arabic linguistic and cultural fidelity represents an important methodological challenge.

Efforts should be directed toward scaling data collection to develop larger and more diverse Arabic multimodal datasets that include additional domains and cultural contexts. Furthermore, translating these research advances into practical applications can enhance accessibility, content management, and educational technologies for Arabic-speaking communities.

The foundations established through ImageEval 2025 provide a robust platform for these future endeavors, with open-source resources and established methodologies enabling continued progress in Arabic multimodal NLP research.

9 Conclusion

ImageEval 2025 represents a significant milestone in Arabic multimodal NLP, addressing the critical gap in Arabic image captioning through innovative task design and community collaboration. The shared task successfully created valuable resources for the research community while highlighting unique challenges and opportunities in Arabic multimodal processing.

Our dual-task approach combining collaborative data creation with competitive model evaluation proved effective in both advancing the state-of-the-art and fostering community engagement. The results demonstrate both the challenges inherent in Arabic image captioning and the potential for significant progress through focused research efforts.

The datasets, evaluation tools, and insights generated through ImageEval 2025 provide a foundation for continued research in Arabic multimodal NLP. We anticipate that this work will catalyze further developments in Arabic vision-language processing and contribute to more inclusive and culturally aware AI systems.

Limitation

This study is limited to Palestinian cultural representation and does not cover other Arabic-speaking regions. The dataset captions are exclusively in MSA and do not include regional dialects. Therefore, while suitable for training models on Palestinian cultural contexts, the dataset's applicability to other Arabic cultures is restricted. Expanding to additional dialects and regions is necessary to enable broader cultural generalization in model training.

Acknowledgments

We thank all participating teams for their valuable contributions to Arabic multimodal NLP. We also acknowledge the organizers of the Arabic NLP 2025 Conference and the broader Arabic NLP community for their support. Special thanks to our annotation quality reviewers and cultural consultants who ensured the authenticity and appropriateness of our guidelines and datasets. This shared task was partially funded by the NPRP14C-0916-210015 grant, awarded by the Qatar Research, Development and Innovation Council (QRDI), and was also partially funded by the Research Committee at Birzeit University (2025T2376).

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. Nadi 2023: The fourth nuanced arabic dialect identification shared task. In *Proceedings of ArabicNLP 2023*, pages 600–613, Singapore (Hybrid). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Amr Keleg, AbdelRahim 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 Abu Horaira, Farhan Amin, Sakibul Hasan, Md. Tanvir Ahammed Shawon, and Muhammad Ibrahim Khan. 2025. Phantomtroupe at imageeval shared task: Multimodal arabic image captioning through translation-based fine-tuning of llm models. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Imad Afyouni, Imtinan Azhar, and Ashraf Elnagar. 2021. Aracap: A hybrid deep learning architecture for arabic image captioning. In *Procedia Computer Science*, pages 382–389.

Diyam Akra, Tymaa Hammouda, and Mustafa Jarrar. 2025. Quranmorph: Morphologically annotated quranic corpus. Technical report, Birzeit University.

Emran Al-Buraihy and Dan Wang. 2024. Enhancing cross-lingual image description: A multimodal approach for semantic relevance and stylistic alignment. *Computers, Materials & Continua*.

Emran Al-Buraihy, Dan Wang, Tariq Hussain, R. Attar, A. Alzubi, Khalid Zaman, and Zengkang Gan. 2025. Aratraditions10k bridging cultures with a comprehensive dataset for enhanced cross lingual image annotation retrieval and tagging. *Scientific Reports*.

Rabee Al-Qasem and Mohannad Hendi. 2025. Impactai at imageeval 2025 shared task: Region-aware transformers for arabic image captioning—a case study on the palestinian narrative. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Mohammed Alkhanafseh, Ola Surakhi, and Abdallah Abedaljalill. 2025. Bzu-aum at imageeval 2025: An arabic image captioning dataset for conflict narratives with human annotation. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

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(8):101684.

Anfal Attai and Ashraf Elnagar. 2020. A survey on arabic image captioning systems using deep learning models. *International Conference on Innovations in Information Technology*.

Israa Al Badarneh, Rana Husni Al Mahmoud, Bassam H. Hammo, and Omar S. Al-Kadi. 2025. Attention-based transformer model for arabic image captioning. *Neural Computing and Applications*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*.

Siddharth Betala and Ishan Chokshi. 2024. Brotherhood at WMT 2024: Leveraging LLM-generated contextual conversations for cross-lingual image captioning. *arXiv preprint arXiv:2409.15052*.

Abdulkadir Shehu Bichi, Ismail Dauda Abubakar, Fatima Muhammad Adam, Aminu Musa, Auwal Umar Ahmed, Abubakar Ibrahim, Khadija Salihu Aua, Aisha Mustapha Ahmed, and Mahmud Said Ahmed. 2025. Codezone research group at imageeval 2025 shared task: Arabic image captioning using BLIP and M2M100—a two-stage translation approach. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Suhyun Cho and Hayoung Oh. 2023. Generalized image captioning for multilingual support. *Applied Sciences*.

Jacob Devlin, Saurabh Gupta, Ross Girshick, Margaret Mitchell, and C. Lawrence Zitnick. 2015. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*.

Samar Elbedwehy and Tamer Mohammed Ibrahim Medhat. 2023. Improved arabic image captioning model using feature concatenation with pre-trained word embedding. *Neural Computing and Applications*, 35:19051–19067.

Passant Elchafei and Amany Fashwan. 2025. Vlcap at imageeval 2025 shared task: Multimodal arabic captioning with interpretable visual concept integration. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

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 (VISAPP)*, pages 233–241.

Jonathan Emami, P. Nugues, A. Elnagar, and Imad Afyouni. 2022a. Arabic image captioning using pre-training of deep bidirectional transformers. *International Conference on Natural Language Generation*.

Jonathan Emami, Pierre Nugues, Ashraf Elnagar, and Imad Afyouni. 2022b. Arabic image captioning using pre-training of deep bidirectional transformers. In *Proceedings of the 15th International Conference on Natural Language Generation*.

Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *European Conference on Computer Vision*, pages 15–29. Springer.

Rana Gaber, Seif Eldin Amgad, Ahmed Sherif Nasri, Mohamed Ibrahim Ragab, and Ensaf Hussein Mohamed. 2025. NU_Internship team at imageeval 2025: From zero-shot to ensembles—enhancing grounded arabic image captioning. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Lianli Gao, Xiangpeng Li, Jingkuan Song, and Heng Tao Shen. 2020. Hierarchical LSTMs with adaptive attention for visual captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:1112–1131.

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*.

Julian Hitschler, Shigehiko Schamoni, and Stefan Riezler. 2016. Multimodal pivots for image caption translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Md. Zakir Hossain, Ferdous Sohel, Mohd. Fairuz Shiratuddin, and Hamid Laga. 2018. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys*, 51:1–36.

Md. Zakir Hossain, Ferdous Sohel, Mohd. Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6):1–36.

George Ibrahim, Rita Ramos, and Yova Kementchedjhieva. 2025. Concap: Seeing beyond english with concepts retrieval-augmented captioning. *arXiv.org*.

Mustafa Jarrar. 2021. The arabic ontology: An arabic wordnet with ontologically clean content. *Applied Ontology*, 16(1):1–26.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa' Omar. 2023a. Wojoodner 2023: The first arabic named entity recognition shared task. In *Proceedings of ArabicNLP 2023*, pages 748–758, Singapore (Hybrid). Association for Computational Linguistics.

Mustafa Jarrar, Nagham Hamad, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2024. Wojoodner 2024: The second arabic named entity recognition shared task. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 847–857, Bangkok, Thailand. Association for Computational Linguistics.

Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2023b. Lisan: Yemeni, iraqi, libyan, and sudanese arabic dialect corpora with morphological annotations. In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7, Egypt. IEEE.

Karima Kadaoui, Hanin Atwany, Hamdan Hamid Al-Ali, Abdelrahman Mohamed, Ali Mekky, Sergei Tilga, Natalia Fedorova, Ekaterina Artemova, Hanan Aldarmaki, and Yova Kementchedjhieva. 2025. Jeem: Vision-language understanding in four arabic dialects. *arXiv.org*.

Mohammed Khalilia, Sanad Malaysha, Reem Suwaileh, Mustafa Jarrar, Alaa Aljabari, Tamer Elsayed, and Imed Zitouni. 2024. Arabicnlu 2024: The first arabic natural language understanding shared task. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 361–371, Bangkok, Thailand. Association for Computational Linguistics.

Nikhil Gopal Khodave and Prathamesh S. Powar. 2025. Survey on multimodal image captioning approaches: Addressing contextual understanding, cross-dataset generalization, and multilingual captioning. *International Journal of Advanced Research in Science, Communication and Technology*.

Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. 2013. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903.

Xirong Li, Chaoxi Xu, Xiaoxu Wang, Weiyu Lan, Zhengxiong Jia, Gang Yang, and Jieping Xu. 2019. COCO-CN for cross-lingual image tagging, captioning, and retrieval. *IEEE Transactions on Multimedia*, 21:2347–2360.

Maofu Liu, Lingjun Li, Huijun Hu, Weili Guan, and Jing Tian. 2020. Image caption generation with dual attention mechanism. *Information Processing & Management*, 57:102178.

Yue Ming, Nannan Hu, Chunxiao Fan, Fan Feng, Jiangwan Zhou, and Hui Yu. 2022. Visuals to text: A comprehensive review on automatic image captioning. *IEEE/CAA Journal of Automatica Sinica*, 9:1339–1365.

Abdelrahman Mohamed, Fakhraddin Alwajih, El Moatez Billah Nagoudi, Alcides Alcoba Inciarte, and M. Abdul-Mageed. 2023a. Violet: A vision-language model for arabic image captioning with gemini decoder. *ARABICNLP*.

Abdelrahman Mohamed, Fakhraddin Alwajih, El Moatez Billah Nagoudi, Alcides Alcoba Inciarte, and Muhammad Abdul-Mageed. 2023b. Violet: A vision-language model for arabic image captioning with gemini decoder. *arXiv preprint arXiv:2311.08844*.

Amal Nayouf, Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian arabic dialects with morphological annotations. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of EMNLP 2023*, pages 12–23, Singapore. Association for Computational Linguistics.

Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. 2023. Improving multimodal datasets with image captioning. *arXiv preprint arXiv:2307.10350*.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2text: Describing images using 1 million captioned photographs. *Advances in Neural Information Processing Systems*, 24.

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. Association for Computational Linguistics.

Hashem Parvin, Ahmad Reza Naghsh-Nilchi, and Hossein Mahvash Mohammadi. 2023. Transformer-based local-global guidance for image captioning. *Expert Systems with Applications*, 223:119774.

Sabri Monaf Sabri. 2021. Arabic image captioning using deep learning with attention. Master's thesis, University of Georgia.

Mariam Saeed, Sarah Elshabrawy, Abdelrahman Hagrass, Mazen Yasser, and Ayman Khalafallah. 2025. Averroes at imageeval 2025 shared task: Advancing arabic image captioning with augmentation and two-stage generation. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Mala Saraswat, Challa Vivekananda Reddy, and Garandal Yashwanth Singh. 2024. Image captioning using NLP. In *2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP)*, pages 549–553. IEEE.

Naeha Sharif, Lyndon White, Mohammed Bennamoun, Wei Liu, and Syed Afaq Ali Shah. 2020. Wembsim: A simple yet effective metric for image captioning. *arXiv*.

Zijie Song, Zhenzhen Hu, Yuanen Zhou, Ye Zhao, Richang Hong, and Meng Wang. 2023. Embedded heterogeneous attention transformer for cross-lingual image captioning. *IEEE Transactions on Multimedia*, 26:9008–9020.

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2023. From show to tell: A survey on deep learning-based image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:539–559.

Akash Verma, Arun Kumar Yadav, Mohit Kumar, and Divakar Yadav. 2023. Automatic image caption generation using deep learning. *Multimedia Tools and Applications*, 83:5309–5325.

Junbo Wang, Wei Wang, Liang Wang, Zhiyong Wang, Dagan Feng, and Tieniu Tan. 2020. Learning visual relationship and context-aware attention for image captioning. *Pattern Recognition*, 98.

Tiantao Xian, Zhixin Li, Canlong Zhang, and Huifang Ma. 2022. Dual global enhanced transformer for image captioning. *Neural Networks*, 148:129–141.

Chenggang Yan, Yiming Hao, Liang Li, Jian Yin, An-An Liu, Zhendong Mao, Zhenyu Chen, and Xingyu Gao. 2021. Task-adaptive attention for image captioning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32:43–51.

Sarah Yassine, Sara Mahrous, and Rawan Sous. 2025. Azlu at imageeval 2025: Bridging linguistic and cultural gaps in arabic image captioning. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Jun Yu, Jing Li, Zhou Yu, and Qingming Huang. 2019. Multimodal transformer with multi-view visual representation for image captioning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30:4467–4480.

Zheng-Jun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. 2022. Context-aware visual policy network for fine-grained image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:710–722.

Jing Zhang, Dan Guo, Xun Yang, Peipei Song, and Meng Wang. 2023. Visual-linguistic-stylistic triple reward for cross-lingual image captioning. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20:1–23.

Qiyuan Zhang, Yufei Wang, Yuxin Jiang, Liangyou Li, Chuhan Wu, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and Chen Ma. 2025. Crowd comparative reasoning: Unlocking comprehensive evaluations for LLM-as-a-judge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5059–5074, Vienna, Austria. Association for Computational Linguistics.

A Methodology

A.1 Zero-Shot Approach

The zero-shot methodology implements direct inference using the pre-trained QWEN2.5-VL-7B-Instruct model without domain-specific fine-tuning, serving as a crucial baseline for Arabic image captioning performance evaluation. The system loads the base model in `bfloat16` precision with automatic device mapping to optimize computational efficiency while maintaining model performance.

A.1.1 Prompt Engineering Strategy

A multimodal prompt was designed combining (i) the raw image and (ii) an instruction asking the model to generate a natural, culturally appropriate caption in Arabic (15–50 words) task within the Palestinian Nakba and Israeli occupation framework:

> *"You are an expert in visual scene understanding and multilingual caption generation. Analyze the content of this image, which is potentially related to the Palestinian Nakba and Israeli occupation of Palestine, and provide a concise and meaningful caption in Arabic - about 15 to 50 words. The caption should reflect the scene's content, emotional context, and should be natural and culturally appropriate. Do not include any English or metadata — The caption must be in Arabic."*

This design leverages the model's pre-trained knowledge about historical events, cultural sensitivity, and multilingual generation capabilities without requiring additional training data.

A.1.2 Inference Pipeline

The system utilizes the processor's chat template functionality for correct input formatting, followed by vision information processing for image data handling. Generation parameters are set with a maximum of 128 new tokens to ensure concise yet meaningful Arabic descriptions while preventing overly verbose outputs.

A.1.3 Methodological Advantages

This zero-shot approach provides several key advantages:

- **Rapid deployment** without training overhead
- **Unbiased evaluation** of pre-trained capabilities
- **Performance baseline** establishment for fine-tuned variant comparison
- **Domain-specific assessment** of cultural sensitivity and historical context understanding in Arabic

The systematic processing and structured CSV output enable comprehensive performance analysis across multiple evaluation metrics, supporting both quantitative assessment through BLEU scores and qualitative evaluation through LLM-as-a-judge scoring systems.

A.2 Fine-tuning Approach

A.2.1 Base Model

We fine-tune `Qwen/Qwen2.5-VL-7B-Instruct`, a vision–language model (VLM) with a unified image encoder and autoregressive text decoder. Supervised fine-tuning (SFT) is performed using LLAMA-FACTORY.

A.2.2 Task Formulation

The objective is Arabic image captioning. Each training example is a two-turn conversation:

- **Human**: "`<image>` Describe this image in Arabic."
- **Assistant**: gold Arabic description.

The dataset template `qwen2_vl` is used so that images and text are tokenized consistently with the base model.

A.2.3 Data Preparation

Source annotations are provided in an Excel file with `File Name` and `Description` columns. We convert rows to the LLaMA-Factory JSON format with absolute image paths:

- `conversations`: the prompt/response pair above.
- `images`: list with one absolute path to the corresponding JPEG.

Before training, we verify the existence and integrity of each image via `PIL.Image.verify()` and report any missing files.

A.2.4 Training Configuration

Parameter-efficient fine-tuning is applied with LoRA:

- **Stage**: SFT **Finetuning**: `lora` on all target modules.
- **LoRA hyperparameters**: rank r=8, α=16, dropout 0.1.
- **Sequence length**: cutoff = 1024 tokens.
- **Batching**: per-device batch size = 1, gradient accumulation = 16 (effective batch size 16).
- **Optimization**: AdamW with learning rate 2×10^{-5}, cosine LR schedule, warmup ratio 0.1.
- **Epochs**: 15.

A.2.5 Logging and Checkpointing

Training is launched via `llamafactory-cli train` with YAML configuration. We log every 5 steps and save checkpoints every 25 steps to the specified output directory. Loss curves are recorded for monitoring; external evaluators are not invoked in this pipeline.

A.2.6 Reproducibility Notes

We confirm tokenizer compatibility with Arabic text and report vocabulary size prior to training. All paths are absolute to avoid path-resolution errors during multi-process loading. The entire procedure is available at https://github.com/SinaLab/ImageCaptionSharedTask2025.

B LLM As a Judge System Prompt

```
You are an expert AI evaluator specializing in Arabic language and semantics. Your task is to act
as an impartial judge and evaluate the quality of a "model-generated caption" of a given image by
comparing it to a "ground truth caption" for the same image. You will not see the image itself.
Your entire evaluation must be based on the textual comparison of the two provided Arabic captions.
Assume the "ground truth caption" is the accurate and correct description of the image.
Evaluation Criteria: Please evaluate the "model-generated caption" based on the following criteria,
using a scale of 1 to 10, where 1 is Very Poor and 10 is Excellent.
Semantic Similarity: - How closely does the model's caption convey the same core meaning as the
ground truth? - Does the caption mention the same key objects, attributes, and actions as the ground
truth? Score 10: The meaning is identical or nearly identical. Score 1: The meaning is completely
different or irrelevant.
REPLY WITH THE SCORE ONLY. NO EXPLANATION
Caption to Evaluate:
```

Codezone Research Group at ImageEval Shared Task: Arabic Image Captioning Using BLIP and M2M100 A Two-Stage Translation Approach for ImageEval 2025

Abdulkadir Shehu Bichi[1], **Ismail Dauda Abubakar**[2], **Fatima Muhammad Adam**[3], **Aminu Musa**[3], **Auwal Umar Ahmed**[4], **Abubakar Ibrahim**[4], **Khadija Salihu Auta**[5], **Aisha Mustapha Ahmed**[6], **Mahmud Said Ahmed**[7]

[1]Baba Ahmed University Kano, [2]Federal University Gusau, [3]Federal University Dutse,
[4]Northbridge College of Science and Technology, [5]Khalifa Isyaku Rabiu University, Kano (KHAIRUN),
[6]Bayero University Kano, [7]Federal University of Technology - Babura

Correspondence: abdulkadir.bichi@babaahmeduniversity.edu.ng

Abstract

This paper details the ImageEval 2025 Shared Task on Arabic image captioning. We designed a two-step, zero-shot framework that utilises the BLIP multimodal vision-language model to first generate English captions. These captions are then converted to Arabic via the M2M100 multilingual translation model. We tested the full pipeline on the official ImageEval 2025 benchmarking set, obtaining a cosine similarity of 0.383 and an LLM Judge score of 15.14. The corroborating numerical and qualitative findings confirm the viability of a translation-driven methodology for cross-lingual image captioning in Arabic, a language often classified as low-resource. Nonetheless, the experiments also uncovered weaknesses: subtle semantic layers and culturally specific references are inadequately conveyed in the output and merit focused attention in subsequent iterations.

Keywords: Arabic image captioning, captioning algorithms, BLIP, M2M100, cross-lingual transfer, multilingual machine translation

1 Introduction

The task of image captioning presents a significant challenge in the fields of computer vision and natural language processing, where models are expected to describe images using natural language. Although considerable effort has been devoted to English image captioning, generating Arabic captions that are both culturally relevant and contextually accurate remains a major challenge due to limited resources and the unique characteristics of the Arabic language (Bashiti et al., 2025).

The ImageEval 2025 Shared Task initiates an Arabic image captioning evaluation framework by providing a dataset of 3,471 images paired with Arabic captions (Bashiti et al., 2025). This initiative aims to facilitate the development of Arabic vision-language models capable of generating culturally relevant and linguistically accurate textual descriptions of images.

Addressing the outlined problem is achievable through a two-step captioning strategy: first, using the leading BLIP model to generate English descriptions of the images, which are then translated into Arabic using the M2M100 multilingual machine translation model. This approach takes advantage of the extensive ImageEval 2025 Shared Task datasets and the strong translation capabilities for the generation of Arabic text. Therefore, our review of the literature suggests that this method is a new technique for Arabic image captioning, since no previous research has used this method to generate captions in Arabic from images.

The remainder of this paper is organized as follows: Section 2 reviews related work, while Section 3 presents our two-stage BLIP+M2M100 methodology. Section 4 describes the experimental setup, followed by Section 5 which presents quantitative and qualitative results with comparative analysis against baseline models. Finally, Section 6 concludes with key findings and discusses future research directions for Arabic image captioning.

2 Related Work

The recent development of vision-language models has predominantly focused on the English language, utilizing models such as CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), and BLIP (Li et al., 2022). These models achieve zero-shot performance by learning joint representations of images and text from large-scale web datasets. However, their application to Arabic remains almost nonexistent, primarily due to the scarcity of available resources and data.

Research on multilingual image captioning has primarily focused on multilingual training (Li et al., 2020), cross-lingual transfer learning (Stefanini et al., 2023), and other translation-based methods (Elliott and de Vries, 2023). Although translation-based approaches to image captioning are straightforward, they are effective when high-quality translation software is available. A significant advancement in multilingual neural machine translation is the M2M100 model (Fan et al., 2021), which can translate directly between one hundred languages without relying on English as a central pivot language.

The processing of Arabic text presents challenges such as complex morphology, diverse dialects, and the right-to-left writing direction (Habash, 2010). These characteristics complicate Arabic text generation and evaluation, underscoring the importance of advancing image-captioning systems in Arabic. This, in turn, fosters the development of Arabic vision-language understanding systems.

3 Methodology

This section explains the system architecture, the process of caption creation for images, translation from English to Arabic, and text normalization.

3.1 System Architecture

The system architecture consists of the following two components.

1. English Caption Generation: The BLIP model generates English descriptions for the given images.

2. Arabic Translation: English captions are translated into Arabic using the M2M100 model. As shown in Figure 1, our proposed system employs a two-stage pipeline approach.

The modular approach enables the utilization of existing English vision-language models alongside state-of-the-art neural machine translation techniques for generating Arabic text.

3.2 Image Captioning and English-to-Arabic Translation

For English captioning, we use the BLIP-base model (Salesforce/blip-image-captioning-base). BLIP employs a unified vision-language pretraining strategy that integrates the training of an image encoder, a text encoder, and an image-grounded text decoder. It is pretrained on large image-text corpora, enabling the model to generate accurate captions for images without prior exposure to specific content. For each provided image, we perform the following steps:

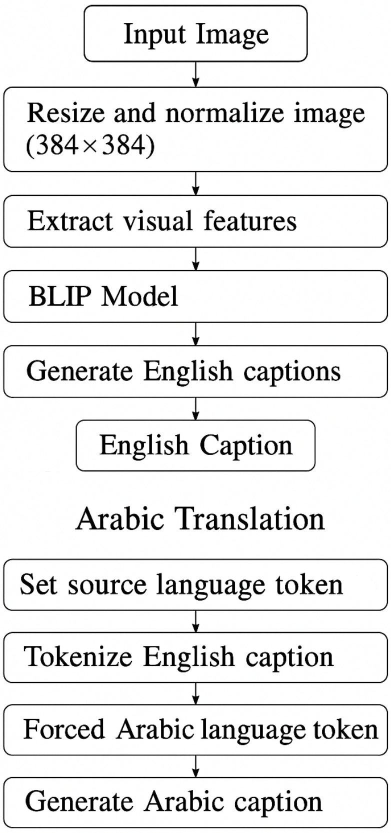


Figure 1: Architecture of the Proposed Model: Arabic Image Captioning Using BLIP and M2M100

1. The image is resized to a resolution of 384 × 384 pixels and then normalized as employed by (Rastogi, 2024).

2. The BLIP vision encoder extracts the corresponding visual features from the image.

3. English captions are generated using beam search decoding.

4. Translation of the caption is performed on the selected caption that has received the highest score.

5. The source language token is set to English (“en”).

6. Tokenization of the English caption using the M2M100 tokenizer.

7. Generation of Arabic translations with the mandatory use of the Arabic language token.

8. Encoding the result to generate the Arabic caption.

3.3 Text Normalization

For evaluation purposes, we performed Arabic text normalization as proposed by (Alami Chehbouni et al., 2020), which includes the following steps: Removing diacritical marks, Removal of Tatweel characters, Removal of punctuation marks, Whitespace Standardization.
This normalization accounts for the morphological intricacies of the Arabic language, providing a robust and fair evaluation against the reference captions.

4 Experimental Setup

This section describes the dataset used, implementation details, and evaluation metrics.

4.1 Dataset

We conducted a system evaluation using the ImageEval 2025 dataset, which comprises 3,471 images with captions, distributed as follows (Bashiti et al., 2025).
1. Training set: 2,718 images
2. Validation set: 75 images
3. Test set: 752 images
The dataset includes a diverse array of visuals accompanied by culturally relevant Arabic captions, making it a robust benchmark for evaluating Arabic image captioning.

4.2 Implementation Details

The following system configuration was used for the implementation:

1. Hardware: Image processing with CUDA-enabled GPUs.
2. BLIP Model: Salesforce/BLIP Image Captioning Based
3. Translation Model: facebook/m2m100-418M
4. Framework: PyTorch with the Transformers library.
5. Inference: Conducted in a zero-shot scenario without any prior model tuning.

The entire processing pipeline generates captions for 752 test images, with an average processing time of 12.27 seconds per image for both caption generation and translation.

4.3 Evaluation Metrics

We used multiple metrics to evaluate the quality of the captions.

1. BLEU Scores: N-gram precision metrics (BLEU-1 through BLEU-4).
2. ROUGE Scores: Recall-oriented metrics including ROUGE-1, ROUGE-2, and ROUGE-L.
3. Cosine Similarity: A metric for evaluating multi-lingual sentence embeddings.
4. LLM Judge Score: Evaluation conducted by large language models.

These metrics analyze various aspects of a caption, including its text, meaning, and human evaluation.

5 Results and Analysis

This section explains qualitative results, error analysis, and provides qualitative examples.

5.1 Quantitative Results

Table 1 reports the metrics achieved by our system in the test set, together with comparisons with the baseline models.

Our dual-pass translation framework achieves significant improvements over standard models, as measured by classical n-gram metrics such as BLEU and ROUGE. In particular, the system achieves substantial gains in BLEU-1 (0.2847 compared to 0.0992 for zero shot and 0.1698 for fine tuned Qwen 2.5-VL), outperforming both zero shot and fine tuned Qwen 2.5-VL models. However, the baseline variants exhibit higher cosine similarity and LLM judge scores, highlighting a complementary balance between precise semantic representation and holistic quality assessment offered by the two architectures.

Table 1: Quantitative Results Comparison – Our Arabic Image Captioning Method vs. Baseline Models

Metric	Our Method (BLIP + M2M100)	Zero-shot Qwen 2.5-VL 7B (Baseline)	Fine-tuned Qwen 2.5-VL 7B (Baseline)
BLEU-1	0.2847	0.0992	0.1698
BLEU-2	0.1623	0.0323	0.0862
BLEU-3	0.0943	0.0190	0.0543
BLEU-4	0.0587	0.0133	0.0305
ROUGE-1	0.0000	0.0000	0.0000
ROUGE-2	0.0000	0.0000	0.0000
ROUGE-L	0.0000	0.0000	0.0000
Cosine Similarity	0.3830	0.5577	0.5846
LLM Judge Score	15.1400	27.1100	30.8200

5.2 Qualitative Assessment Outcomes

Beyond numerical evaluation, the framework was assessed using qualitative criteria that addressed both the cultural and linguistic appropriateness of the generated captions.

1. Cultural Relevance: 1.10
2. Conciseness: 2.03
3. Completeness: 1.47
4. Accuracy: 2.03

The modest scores highlight a pressing need to improve cultural depth and caption comprehensiveness, while conciseness and accuracy demonstrate steady, if not outstanding, performance.

5.3 Error Analysis

This section, examining the generated captions, uncovers some recurring issues.

1. Idioms: Some English phrases and sayings do not have an equivalent in Arabic.
2. Cultural Relevance: Some generated captions include references that lack culturally relevant details or specific information. Morphological variations in Arabic pose challenges to exact lexical matching due to its complex morphology.
3. Object Misrecognition: Some devices or ideas that belong to specific cultures are misrecognized.

5.4 Comparative Analysis

Our two-stage approach demonstrates distinct performance characteristics compared to the baseline models.

Strengths:

1. Led performance evaluation using surface n-gram overlap metrics (BLEU, ROUGE).
2. Leveraging advanced English vision-language encoding techniques
3. Stable pipeline extending from Arabic vision to generated captions

Limitations:

1. Achieved through dedicated multilingual captioning models.
2. Performance on large language model evaluation metrics continues to lag behind state-of-the-art benchmarks.
3. Qualitative assessments identify instance-specific gaps in cultural relevance, reaffirming the necessity of localized context.

However, results indicate that translation-based pipelines produce captions with high linguistic fidelity to reference standards, whereas models that retain multilingual embeddings convey deeper semantic information, albeit with slightly lower lexical precision.

6 Limitation

No Direct Visual-Arabic Learning: The approach cannot learn how Arabic speakers naturally describe visual content, relying instead on English visual understanding followed by translation, which misses Arabic-specific visual-linguistic patterns.

7 Conclusion and Future Work

The shift and contribution of this paper lies within the use of the BLIP and M2M100 to produce a new two-stage Arabic image captioning system as well as the comparative Qwen 2.5-VL (zero-shot and fine-tuned) baseline analysis of the Arabic datasets. The regional context has not previously been analyzed.

Furthermore, this study presents a two-phase Arabic captioning architecture that takes advantage of existing English vision-language models alongside tile-based multilingual translation services. The resulting system achieved a competitive cosine similarity score of 0.383, demonstrating the feasibility of translation-centric cross-lingual image captioning. Performance metrics indicate that translation-based approaches outperform conventional n-gram baselines; however, evaluations of semantic coherence and cultural representation reveal gaps that require targeted refinement. Future iterations will incorporate deeper multilingual embeddings and culturally aware context modules to enhance both meaning preservation and cultural resonance.

Future enhancements may include the following:

1. Direct Arabic Vision-Language Models: Develop fully end-to-end Arabic image captioning systems that utilize large-scale, culturally specific datasets to maximize relevance.
2. Cultural Context Enhancement: Integrate structured cultural knowledge graphs with annotation pipelines to ensure that relevance scoring and caption generation reflect nuanced local traditions.
3. Hybrid Approaches: Combine the rigor of lexically precise, translation-inspired systems with the deep semantic capabilities of multilingual transformers within a balanced,

modular, and selective architecture.

4. Advanced Text Normalization: Implement state-of-the-art morphological disambiguation and dialect-aware normalization techniques to standardize Arabic text while minimizing contextual distortion.

The ongoing research highlights the rapid progress in Arabic artificial intelligence. The upcoming ImageEval 2025 benchmark is expected to further intensify competition in Arabic vision and language comprehension.

Acknowledgments

The ImageEval 2025 organization has provided a detailed evaluation and dataset that have contributed significantly to the advancement of Arabic artificial intelligence. The creators of the dataset and evaluation framework have offered invaluable resources for Arabic Natural Language Processing.

Code Availability Statement

The code supporting the findings of this study is openly available at `https://github.com/asbichi362/Arabic-Image-Captioning-Using-BLIP-and-M2M100`

References

A. Alami Chehbouni, A. Ouatik Said, and T. Rachidi. 2020. A proposed natural language processing preprocessing procedures for enhancing arabic text summarization. In *Advances in Intelligent Systems and Computing*, pages 25–34. Springer.

Ahlam Bashiti, Alaa Aljabari, Hadi Hamoud, Md. Rafiul Biswas, Bilal Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouani. 2025. ImageEval 2025: The First Arabic Image Captioning Shared Task. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

J. Elliott and A. P. de Vries. 2023. Evaluating image captioning systems via cross-modal retrieval. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12543–12559.

A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, N. Goyal, T. Birch, V. Liptchinsky, S. Edunov, E. Grave, M. Auli, and A. Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

N. Y. Habash. 2010. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. *Proceedings of the 38th International Conference on Machine Learning*, pages 4904–4916.

J. Li, D. Li, C. Xiong, and S. Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Proceedings of the 39th International Conference on Machine Learning*, pages 12888–12900.

X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 121–137.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, G. Krueger J. Clark, and I. Sutskever. 2021. Learning transferable visual models from natural language supervision. *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763.

Ritvik Rastogi. 2024. Papers explained 190: Blip-3 (xgen-mm). `https://ritvik19.medium.com/papers-explained-190-blip-3-xgen-mm-6a9c04a3892d`. Medium.

M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, and R. Cucchiara. 2023. From show to tell: A survey on deep learning-based image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):539–559.

BZU-AUM at ImageEval 2025 Shared Task: An Arabic Image Captioning Dataset for Conflict Narratives with Human Annotation

Mohammed AlKhanafseh
Birzeit University
malkhanafseh@birzeit.edu

Ola Surakhi
American University of Madaba
o.surakhi@aum.edu.jo

Abdallah Abedaljalill
Birzeit University
1212725@birzeit.edu

Abstract

This paper presents a new Arabic image captioning dataset created for the ImageEval 2025 Shared Task. The dataset focuses on images related to conflict, resistance, and everyday life under occupation. Each image is paired with a Modern Standard Arabic caption of 40–70 words that describes what is shown and adds cultural or emotional context. To help annotators write rich and consistent captions, we used prompt-based guidelines, including step-by-step reasoning and writing from specific roles such as journalists or humanitarian observers. This method produced captions that are both descriptive and meaningful. The dataset fills an important gap in Arabic resources, especially for sensitive and historically significant topics. It can be used to train and evaluate Arabic vision language models, test multilingual AI systems, and support applications in journalism, education, and cultural preservation.

1 Introduction

The ImageEval 2025 Shared Task encourages multilingual, culturally sensitive image captioning by having participants create Arabic captions for pictures that call for emotional, historical, or cultural knowledge. We make a contribution by compiling a dataset of Arabic captions that have been human-annotated for photos that show resistance, conflict, and day-to-day life under occupation. A crucial task in computer vision and natural language processing, image captioning produces natural language descriptions for visual content and facilitates uses like content indexing, accessibility, and visual storytelling. Even though the field has been advanced by large datasets like MS-COCO citelin2014microsoft, Flickr30k citeyou2016image, and Visual Genome citekrishna2017visual, they limit culturally rich or emotionally charged narratives by concentrating on English and generic domains. Despite being widely spoken, Arabic is still under-represented, which limits the use of large language models and vision-language in these situations. With thorough, human-written captions in formal Arabic (40–70 words) that capture both the visible content and any underlying cultural or emotional meaning, our dataset fills this gap. We used Chain-of-Thought (Wei et al., 2022; Kharma et al., 2025) and Role-Based prompting (Bubeck et al., 2023) to guarantee consistency and depth, directing annotators to reason methodically and adopt viewpoints similar to those of a journalist or witness. This tool facilitates the development of socially conscious AI systems, the training and assessment of Arabic vision-language models, and the improvement of multilingual LLMs. Additionally, it offers a standard for the ImageEval 2025 Shared Task, allowing models to be assessed on culturally relevant and contextually rich captions.

In the remaining portion of the paper, relevant work, dataset construction, annotation methodology, caption analysis, use cases, and future directions are discussed.

2 System Overview

In this section, we describe the dataset creation and annotation process used in our submission to the ImageEval 2025 Shared Task.

2.1 Dataset Composition and Historical Context

With an emphasis on the Palestinian experience as it has been shaped by resistance, displacement, and colonisation, this dataset tackles the dearth of culturally rich and emotionally charged Arabic image captioning data. In contrast to other datasets that contain neutral, apolitical content, it contains images of life under occupation, acts of protest and survival, and aspects of cultural continuity and defiance. These images function as digital repositories of memory and identity in addition to being training

Proceedings of The Third Arabic Natural Language Processing Conference, pages 395–400
November 8-9, 2025 ©2025 Association for Computational Linguistics

data for vision-language models.

Two datasets of 250 images each, arranged into five thematically distinct sheets, were selected for analysis. As outlined in Table 1, each sheet focusses on a distinct facet of Palestinian history and daily life, guaranteeing coverage of both traumatic and resilient experiences.

Our dataset's imagery captures the nuanced historical and sociopolitical background of Palestine. Over 700,000 Palestinians were displaced and over 400 villages were destroyed during the 1948 Nakba, which was the result of political repression, house demolitions, military raids, and growing Zionist settlement during the British Mandate (1917–1948). Images of destroyed homes, military checkpoints, refugee camps, civilian resistance, and everyday resiliency were produced by later events, such as the 1967 Six-Day War and the ongoing occupation of the West Bank, Gaza, and East Jerusalem. This range of adversity and resilience is reflected in the photographs we have chosen. Archival collections, documentary photography, and publicly accessible materials that adhere to ethical and legal guidelines are examples of sources. While avoiding exploitative or dehumanising content, each image was carefully assessed for its emotional and historical significance. The captions emphasise social cohesion, cultural pride, and dignity while highlighting both suffering and resiliency.

In addition to offering top-notch, ethically sourced content for training and assessing Arabic vision-language models, this curation approach guarantees that the dataset portrays a complex, multi-layered Palestinian narrative that is frequently missing from widely used computer vision datasets.

2.2 Prompt-Guided Annotation Strategy

We used a structured prompt engineering approach to generate high-quality captions for the shared task, allowing for expressive captions that go beyond straightforward image descriptions. Formal captions of 40–70 words per image were written by native Arabic speakers who had received training in descriptive writing and sociopolitical context.

There are two primary methods for guided annotation: Role-Based prompting (Bubeck et al., 2023) and Chain-of-Thought prompting (Wei et al., 2022; Kharma et al., 2025). Methodical reasoning was promoted by Chain-of-Thought prompting, in which annotators first discussed observable elements (people, objects, and setting), then thought about actions or events, and lastly discussed historical, symbolic, or emotional ramifications. A child standing next to debris, for instance, could be explained not only in terms of the obvious damage but also in terms of the larger context of displacement or societal memory. As a result, the captions were contextually rich and semantically layered. Annotators were given distinct viewpoints, such as journalist, eyewitness, or humanitarian, along with suggested questions and tones for each role, thanks to role-based prompting. This method maintained thematic coherence while expanding the emotional and stylistic scope of captions.

Grammar, clarity, cultural correctness, and the absence of bias or conjecture were all examined in each caption. When an image had more than one caption, narrative and emotional significance were given priority during the selection process.

This dataset creates insightful, culturally relevant captions and allows for deeper interaction with images through the integration of structured prompting. It offers a standard for assessing Arabic vision-language models on content that demands both factual accuracy and narrative depth, and it methodologically advances socially conscious AI.

2.3 Experimental Setup

Data. We use 500 images with one caption each, organized in two batches (250/250). Each row has image_id, caption, and batch_id. Text is UTF-8 and in Modern Standard Arabic (MSA).

Annotation. Native Arabic speakers wrote the captions in a spreadsheet interface. They followed the template in Section 3.2 (role = journalist, eyewitness, or humanitarian). Rules: 40–70 words, MSA only, no speculation, no identification of minors.

Review. We used a two-pass review. Checks covered length, grammar, MSA register, factual grounding, tone, and role. Items that failed were corrected or replaced.

Stats. The numbers in Section 4 were computed with a simple script: word count by whitespace tokens and sentence count by Arabic/English punctuation.

Packaging. We provide a CSV with image_id, caption, batch_id. Prompt texts and scripts will be released after review.

3 Results and Analysis

This section reports the caption characteristics used in our ImageEval 2025 shared-task submission.

3.1 Linguistic and Structural Characteristics of the Captions

For ease of clarity, formality, and cultural depth, all of the captions are written in Modern Standard Arabic (MSA); the use of colloquial language was avoided given that MSA is preferred for academic, journalistic, and historical concerns. All captions are around 50 words in length, but could range from 15 words to 100 words in length. This is enough room to describe the visual content, while also adding context that was cultural, historical, and emotional. Captions are often more than facts, they represent a thoughtful and interwoven narrative that conveys historical significance and meaning.

Using formal, understandable language, this caption opens with an objective, detailed description of the visual scene, emphasising the subject's posture, surroundings, and companions. It then deciphers traditional attire and facial expressions as symbolic expressions of resistance and identity. Lastly, it highlights the image's political and cultural significance by placing it within a larger historical narrative. Reflecting the Chain-of-Thought prompting technique employed during annotation, the structure logically moves from particular visual details to general historical significance.
Similarly, foe example this caption presents a historic landmark with rich cultural connotations:
"تظهر هذه الصورة التاريخية قبة الصخرة المشرفة في القدس، بقبتها الذهبية المميزة وعمارتها الإسلامية الأصيلة، محاطة بأشجار السرو الشامخة والساحات الرحبة للحرم الشريف. تمثل هذه اللقطة النادرة من أوائل القرن العشرين جمال المسجد الأقصى وقداسته، حيث يقف هذا المعلم الإسلامي شاهداً على تاريخ فلسطين العريق وحضارتها الإسلامية. تجسد الصورة الهدوء والسكينة التي تخيم على هذا المكان المقدس، الذي يحمل في طياته ذكريات الإسراء والمعراج ومكانة القدس الخاصة في قلوب المسلمين في جميع أنحاء العالم."
Here, the caption highlights the landmark's religious and cultural significance while creatively describing the surrounding landscape and architectural beauty. Words like "القداسة" and "السكينه", arouse the reader's emotions and spirituality, urging them to value the location beyond its outward appearance. The caption maintains coherence and narrative flow in accordance with the dataset's guidelines by fusing together visual, historical, and cultural layers.
A third example depicts a historically significant landscape:
"تُظهر هذه الصورة التاريخية النادرة جبل الزيتون من باب النبي داوود في القدس، حيث تمتد التلال المقدسة والوديان التي شهدت أحداثاً تاريخية ودينية عظيمة عبر القرون. يبدو في المشهد جزء من أسوار القدس القديمة الحجرية والمباني التراثية المتناثرة على سفوح الجبل، بينما تظهر في الأفق مئذنة تشير إلى الحضور الإسلامي العريق في المدينة المقدسة. تحكي هذه اللقطة من أوائل القرن العشرين قصة القدس الخالدة، بتضاريسها الوعرة وتاريخها المتجذر في قلب فلسطين، مجسدة الطابع الروحاني والثقافي الذي يميز هذه الأرض المباركة."
This caption incorporates detailed topographical description, historical references, and spiritual symbolism, capturing the multifaceted nature of the depicted scene. The use of descriptive phrases like التلال المقدسة and ""التاريخ المتجذر reflects an elevated linguistic style that enriches the viewer's understanding. The caption also conveys a temporal dimension by situating the image historically "من أوائل القرن العشرين", enhancing its documentary value.
All of the captions in this dataset have a similar structure, in that they describe the visible components in the picture first, and then provide an interpretation of the image based on a historical, cultural, and/or emotional context, in order to introduce meaning to the submissions to train vision-language models. Role-based prompting provides a way to translate the different perspectives: witness or humanitarian perspectives use emotional, personal language, whereas journalistic captions sought objectivity and clarity. Having multiple voices is important to allow for multiple perspectives of conflict, identity, and cultural heritage to contribute to a cohesive narrative. The dataset with the relatively balanced length of captions, formality of Modern Standard Arabic, and content that is culturally meaningful offers the opportunity to contribute to the understanding of Arabic vision-language models in sensitive and complex domains.

3.2 Dataset-level

Table 3 shows the dataset is length-controlled yet varied: mean ≈54 words (median 51), 79.2% within the 40–70 target, and about 2.7 sentences per caption.

3.3 Shared Task Evaluation Results

With a BLEU-4 score of 0.41 and a ROUGE-L score of 0.56, our system demonstrated a high degree of phrasing and structural alignment with reference captions. Additionally, it received an LLM judge score of 32.42 and a cosine similarity mean of 65.53, which indicate linguistic fluency and semantic closeness. According to these findings, our system outperformed all others in terms of meaning accuracy and fluency. Only a few minor visual details were left out of the captions, which did a good job of capturing the cultural and historical context.

4 Use Cases and Future Directions

This Arabic image captioning dataset, emphasizing cultural identity, resistance, and conflict, enables research on vision-language models and the development of socially significant AI.

4.1 Use Cases

This dataset pairs historically significant images with culturally rich captions to support a variety of important applications:

- **Multimodal LLM and Arabic VLM training:** Improves models to comprehend intricate historical, cultural, and affective image contexts.
- **Assistive technologies:** Enhances accessibility for Arabic speakers by offering more detailed, contextualised descriptions of images..
- **Cultural heritage preservation:** Aids in recording and disseminating Palestinian history and regional conflicts to educational institutions.
- **Digital archiving:** Makes it possible to create searchable, semantically rich archives that aid in research and preserve collective memory.
- **Journalism and humanitarian work:** Automates fact-checking of photos from conflict areas and sensitive, accurate storytelling.

4.2 Future Directions

Although this dataset provides a useful tool for researching the relationship between language, culture, and history, much more could be done to expand its use, boost its influence, and guarantee responsible use. The following areas will be the focus of future efforts:

- Increase the number and variety of images.
- Incorporate more languages and a greater variety of Arabic dialects.
- Examine how language models handle bias, cultural quirks, and emotions using the dataset.

5 Limitations and Ethical Considerations

Even though this dataset has special historical and cultural significance, responsible use requires acknowledging its limitations and ethical issues. These elements are summed up in the points that follow:

Limitations:

- **Small size:** Only 250 images, limiting diversity of visuals and contexts (Torralba and Efros, 2011).
- **Language:** Captions in Modern Standard Arabic (MSA) ensure consistency but exclude dialectal nuances (Abdul-Mageed et al., 2023).
- **Generalizability:** Integration with other datasets is recommended (Nguyen and Ploeger, 2025).

Ethical Considerations:

- Includes delicate and possibly upsetting material (conflict, trauma, and displacement).
- Risks of abuse, deception, or retraumatization.
- Guidelines for annotations placed a strong emphasis on factual, courteous descriptions that shunned bias or sensationalism.

6 Conclusion

The lived experiences of Arabic-speaking communities impacted by historical trauma and conflict are being connected to AI for the first time with this dataset. It does more than just describe pictures; it tells stories with social and emotional significance, highlighting cultural heritage, resiliency, and resistance. AI can now interpret images while honouring the voices and histories they represent thanks to this method. To guarantee that future AI tools continue to be morally and culturally appropriate, we encourage continued cooperation between linguists, historians, AI researchers, and local communities. The AI community can support social justice and cultural memory preservation by growing and improving these datasets and encouraging inclusive practices.

In the end, this resource shows how AI can be used not only for Arabic vision-language research but also as a tool for empathy, comprehension, and historical preservation, encouraging work that respects human experience and crosses cultural divides.

Appendix: Additional Tables

Table 1: Overview of key historical events and figures in Palestinian and Rif resistance history.

Topic	Details
Mohammed bin Abdelkrim El Khattabi El Ouryaghli	Moroccan judge and fighter, leader of the Rif resistance against Spanish and French colonialism, founder of the Republic of the Rif (1882–1963).
1948 War	Started May 15, 1948; Arab-Zionist conflict caused mass Palestinian displacement ("Nakba").
Jerusalem	Capital of Palestine, historic and religious city with Al-Aqsa Mosque; strategic central highlands location.
Palestinian Cities Occupied in 1948	Haifa, Acre, Jaffa: cultural and commercial hubs; Nablus and Bethlehem: religious and historical significance.
British and Zionist Colonialism (1917–1948)	Palestinians faced repression and displacement during the British Mandate, resisted through uprisings, strikes, and armed struggle.
Zionist Attacks on Beirut	Israeli invasion caused widespread destruction and thousands of civilian casualties.
Lebanese Civil War (1975–1990)	Fifteen-year multifaceted conflict with 120,000 deaths and millions displaced, involving multiple sectarian, Palestinian, Israeli, Syrian, and international actors.

Table 2: Summary of key historical events and cultural aspects of Palestine and the Rif region (second dataset).

Topic	Details
British and Zionist Colonialism (1917–1948)	Persecution escalated under the British Mandate, including home demolitions, military raids, and displacement.
1967 War	1967 war caused Israeli occupation of key territories and further Palestinian displacement.
Jerusalem	Capital of Palestine; historic city with Al-Aqsa Mosque and Dome of the Rock, Islam's third holiest site
Palestinian Cities Occupied in 1948	Haifa, Acre, Jaffa: historic trade centers; Nablus, Bethlehem: key religious and cultural sites.
Daily Life in Palestine	Markets, religious centers, rural herding, fishing, and glassmaking reflect social and economic diversity.

Table 3: Caption statistics (overall).

Statistic	Value
Images	500
Captions	500
Target length (words)	40–70
Mean words per caption	53.99
Median words per caption	51
Sentences per caption (avg.)	2.72
% within 40–70 words	79.2%
Min / Max words	3 / 148

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. Nadi 2023: The fourth nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2310.16117*.

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Ahlam Bashiti, Alaa Aljabari, Hadi Hamoud, Md. Rafiul Biswas, Bilal Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouani. 2025. ImageEval 2025: The First Arabic Image Captioning Shared Task. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Jiaxin Ge, Hongyin Luo, Siyuan Qian, Yulu Gan, Jie Fu, and Shanghang Zhang. 2023. Chain of thought prompt tuning in vision language models. *arXiv preprint arXiv:2304.07919*.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mohammed Kharma, Soohyeon Choi, Mohammed AlKhanafseh, and David Mohaisen. 2025. Security and quality in llm-generated code: A multi-language, multi-model analysis. *arXiv preprint arXiv:2502.01853*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, and 1 others. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Dong Nguyen and Esther Ploeger. 2025. We need to measure data diversity in nlp–better and broader. *arXiv preprint arXiv:2505.20264*.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Daniel Rose, Vaishnavi Himakunthala, Andy Ouyang, Ryan He, Alex Mei, Yujie Lu, Michael Saxon, Chinmay Sonar, Diba Mirza, and William Yang Wang. 2023. Visual chain of thought: bridging logical gaps with multimodal infillings. *arXiv preprint arXiv:2305.02317*.

Antonio Torralba and Alexei A Efros. 2011. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE.

Ning Wang, Jiahao Xie, Jihao Wu, Mingbo Jia, and Linlin Li. 2023. Controllable image captioning via prompting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2617–2625.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659.

Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. 2023. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191.

ImpactAi at ImageEval 2025 Shared Task: Region-Aware Transformers for Arabic Image Captioning – A Case Study on the Palestinian Narrative

Rabee Al-Qasem
AI Developer, GGateway
Nablus, Palestine
r.qasim@ggateway.tech

Mohannad Hendi
Independent Researcher
Nablus, Palestine
M.hendi1@student.aaup.edu

Abstract

This paper presents our approach to the ImageEval Shared Task for Arabic image captioning, with a focus on the Captioning with Region Features Transformer (CRAFT) model. The system combines Faster R-CNN-based region feature extraction with a custom vision transformer encoder and transformer decoder, trained on a custom, human-annotated dataset with a Palestinian context. To ensure fairness in evaluation, we compare CRAFT with an alternative Vision-Encoder–Decoder system (AraViT-GPT). Performance was assessed using BLEU, ROUGE, cosine similarity, and an LLM-based semantic evaluation. Results show that CRAFT achieved the highest cosine similarity (56.22 on the test set), indicating superior semantic fidelity to reference captions, while AraViT-GPT showed marginally better n-gram precision and LLM-judge scores. These findings demonstrate the advantages of region-focused visual encoding for Arabic caption generation, particularly in the context of context-rich and historically significant imagery.

1 Introduction

This paper presents our work in **Subtask 2 of the ImageEval 2025 Shared Task** on developing and evaluating image captioning models (Bashiti et al., 2025). This subtask focuses on generating culturally relevant and contextually accurate Arabic captions for images.

We developed **CRAFT** (Captioning with Region Features Transformer), which uses region-level visual features extracted via Faster R-CNN (ResNet-50 backbone), followed by a custom vision transformer encoder and transformer decoder. We also compared this main model with a custom transformer-based model, **AraViT-GPT**, as well as the baseline model provided in the shared task.

Our experiments showed that CRAFT achieved the highest semantic fidelity, with cosine similarity scores of 57.22 (validation) and 56.22 (test), while AraViT-GPT slightly outperformed in n-gram precision and LLM judge scores. In the official leaderboard, our system ranked 4th in both cosine similarity and LLM-based evaluation, and 5th in the human evaluation track, where real annotators assessed caption quality.

The main challenge we encountered was named entity recognition, where the model occasionally produced factual inaccuracies when identifying specific people or locations, despite correctly recognizing the general scene context. Our code and training pipeline are available at Github .

2 Background

The main task addressed in this paper is image captioning (Subtask 2), where the model takes an image as input and generates an Arabic caption for it. The model's performance is then evaluated using metrics such as BLEU, ROUGE, cosine similarity, and an LLM-based semantic scoring metric.

The provided dataset consists of 3,471 manually captioned images in Arabic. The dataset encompasses a diverse range of scenes, including buildings, people, and artifacts. It presents various challenges, such as identifying individuals' names and handling both colored and black-and-white images. Each image features a unique Arabic caption, annotated by a human, that provides a detailed description of the image. The dataset particularly focuses on the Palestinian historical narrative.

3 System Overview

Our empirical study compares two algorithms: Captioning with Region Features Transformer (CRAFT) and (AraViT-GPT). In this section, we discuss the CRAFT model, which achieved the highest cosine similarity results of the two models. Details of AraViT-GPT are in Section 6, which we included as part of our ablation study.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 401–407
November 8-9, 2025 ©2025 Association for Computational Linguistics

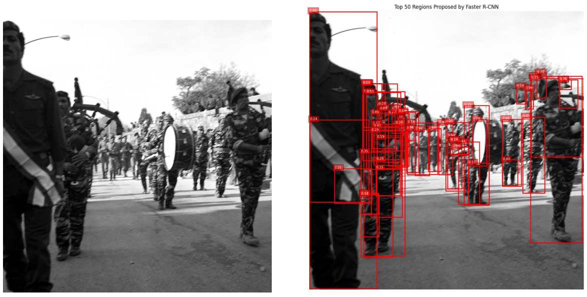


Figure 1: Faster R-CNN output showing region proposals, capturing multiple people and contextual objects.

3.1 Region Feature Extraction using Faster R-CNN

Given the heterogeneity of our images (crowds, many objects, and photo-of-photo artifacts), we use a pre-trained Faster R-CNN (Ren et al., 2015) with a ResNet-50 backbone (He et al., 2016) to propose regions. Rather than process full images, we extract $k = 50$ regions of interest (ROIs) per image, yielding a 50×1024 embedding tensor; these features, together with normalized box coordinates, serve as input to the vision transformer backbone (Fig. 1). We also tested a dynamic k chosen by clustering ROIs via the elbow method (typically 15–70 per image), but it sometimes dropped important objects (Hendi et al., 2023). Also we tried a fixed $k = 100$, which offered no improvement over $k = 50$.

3.2 Visual Encoder

Our custom Vision Encoder processes region features from Faster R-CNN. The features are first unified by a projection layer, combined with learned positional embeddings, and then passed to a Transformer encoder with an optimal configuration of two layers (L=2) and two attention heads (H=2), as determined by hyperparameter tuning using a grid search (Bergstra and Bengio, 2012). This process generates a [Batch, 50, 768] embedding that is subsequently passed to the caption decoder (Fig. 2).

3.3 Caption Decoder

The caption decoder generates Arabic text using ArabGlossBERT (Al-Hajj and Jarrar, 2022; Antoun et al., 2020), which provides a vocabulary of approximately 64,000 tokens. Token and positional embeddings are mapped to a 768-dimensional space to match the visual features from the encoder. Both text embeddings and visual features are then fed into a Transformer decoder. This decoder is configured with a maximum caption length of M=97, with its optimal parameters of L=2 layers and H=2 attention heads determined by a grid search (Fig. 3).

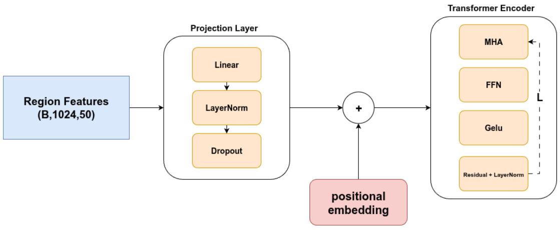


Figure 2: Architecture of the Vision Encoder, showing the three main components: projection of region features, addition of learned positional embeddings, and a multi-layer Transformer encoder.

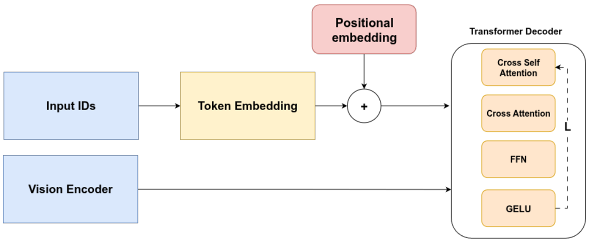


Figure 3: Architecture of the Caption Decoder, showing token embedding, addition of positional embeddings, Transformer decoder, and final projection to vocabulary.

3.4 Sequence Decoder

For caption generation, we use the beam search method (Vaswani et al., 2017). In this iterative process, the decoder maintains a set of candidate sequences (beams) at each step. A causal mask is applied to prevent the model from attending to future positions, and only the top candidates, ranked by their cumulative probabilities, are retained. The process terminates upon reaching the maximum sequence length of 97, at which point the highest-scoring sequence is selected as the final caption.

4 Experimental Setup

4.1 Data Split

All experiments were conducted using the Shared Task dataset, with no external data involved in the process. The dataset was split into training (2,718 samples) and test (753 samples) sets. During the Task, we also received 75 images for validation. The test set was used exclusively to evaluate the models' generalization performance.

4.2 Data Preprocessing

We normalized all Arabic captions to reduce noise: unified orthographic variants (Alef, Yeh, Teh Marbuta), removed Tatweel and diacritics, standardized punctuation to Latin equivalents, and collapsed redundant whitespace. The slash (/) in date-like captions was treated as an unknown token by the tokenizer, so we replaced it with a space (e.g., ‘09/1970’ → ‘09 1970’) before tokenization. These steps improved the model’s predictions.

4.3 Training setup

Prior to the full training, we conducted hyperparameter tuning over a limited run of 10 epochs. The grid search included the number of encoder and decoder layers, the number of attention heads, and the batch size. For more details on the hyperparameter tuning, see Appendix A.1. The optimal configuration was found to be 2 layers for both the encoder and decoder, 2 attention heads, and a batch size of 8 (see Appendix A.1). This resulted in a model size of approximately 132 million trainable parameters.

Using these settings, the full training was performed on a NVIDIA T4 GPU via Google Colab for 40 epochs with early stopping, which occurred at epoch 30. The AdamW optimizer (Loshchilov and Hutter, 2017) was employed alongside a linear learning rate scheduler initialized at 1×10^{-4}. Cross-entropy loss was selected as the objective function. To further enhance model generalization, we applied online data augmentation where each sample was exposed to randomized transformations on every epoch. These included horizontal flips, mild affine transformations (rotations up to $\pm 15^\circ$, translations up to ± 5, scaling between 0.9–1.1, and shear up to $\pm 3^\circ$), as well as photometric changes such as brightness and contrast adjustments, gamma correction to simulate aging effects, and the addition of light Gaussian noise with $\sigma \leq 0.05$. Collectively, these augmentations increased sample diversity and made the model more robust to variations in historical images.

Finally, our implementation, developed in Python, utilized PyTorch and TorchVision for model training (Paszke et al., 2019; Marcel and Rodriguez, 2010), alongside NumPy. We used Matplotlib and Seaborn for visualization, Hugging Face Transformers for transformer components, and Weights & Biases (W&B) for experiment tracking.

4.4 Evaluation Metrics

To comprehensively assess both models, we used a mixed set of evaluation metrics provided in the shared task description paper (Bashiti et al., 2025): BLEU (Papineni et al., 2002) for n-gram precision, ROUGE (Lin, 2004) for recall-oriented overlap, cosine similarity for semantic alignment in embedding space, and a Large Language Model (LLM) judge (GPT-4o) to imitate human judgments (Al-Qasem et al., 2025). While BLEU and ROUGE quantify surface overlap, cosine similarity indicates whether a predicted caption conveys the reference meaning. Because semantic similarity is our primary objective, we assign greater weight to cosine similarity when comparing models. This weighting, together with the LLM judge, guided our conclusion about which model best suits the target application.

5 Results

In this section, we report the performance of the two models, provide examples from the best-performing model, and highlight some flaws observed in its predictions. As shown in Table 1, and following our protocol in Section 4, cosine similarity is treated as the *primary* metric because it best captures semantic fidelity to the reference captions.

5.1 Quantitative findings

The results in Table 1 show that the CRAFT model achieves higher scores on both splits in terms of cosine similarity, with 57.22 on the validation set and 56.22 on the test set, indicating greater semantic closeness in the embedding space. On the other hand, AraViT-GPT holds a slight lead in n-gram precision (BLEU-1–4) and achieves the highest LLM-Judge score (26.55 on the test set). Both models record near-zero results on the ROUGE metrics, which in this case reflects the lack of exact lexical overlap between the generated and reference captions. For context, we also benchmarked both models against the shared-task baseline, see Table 2. While both models exceed the baseline on BLEU-1–4, the fine-tuned baseline attains the highest cosine similarity (58.46) and LLM-as-judge score (30.82) on the test set.

5.2 Prediction Discussion

To complement the quantitative results, we present a qualitative comparison between the two models’ outputs on selected images from the Test dataset.

Metric	CRAFT		AraViT-GPT	
	Val	Test	Val	Test
BLEU-1	19.68	19.07	**21.05**	**21.40**
BLEU-2	10.56	9.66	**11.48**	**11.59**
BLEU-3	7.01	6.00	**8.54**	**8.48**
BLEU-4	4.21	3.78	**5.18**	**5.22**
ROUGE-1	0	0	0	0
ROUGE-2	0	0	0	0
ROUGE-L	0	0	0	0
Cosine Similarity Mean	**57.22**	**56.22**	55.35	55.46
LLM Judge (/100)	22.07	22.34	**26.07**	**26.55**

Table 1: Comparison of CRAFT and AraViT-GPT performance on validation and testing sets.

	Ours		Baseline (Qwen 2.5-VL 7B)	
Metric	CRAFT (Test)	AraViT-GPT (Test)	Zero-shot	Fine-tuned
BLEU-1	19.07	**21.40**	9.92	16.98
BLEU-2	9.66	**11.59**	3.23	8.62
BLEU-3	6.00	**8.48**	1.90	5.43
BLEU-4	3.78	**5.22**	1.33	3.05
ROUGE-1	0	0	0	0
ROUGE-2	0	0	0	0
ROUGE-L	0	0	0	0
Cosine Similarity Mean	56.22	55.46	55.77	**58.46**
LLM Judge (/100)	22.34	26.55	27.11	**30.82**

Table 2: Test-set comparison between our models and the baseline. Baseline values are taken from the shared-task notebooks and converted to a percentage scale for comparability.

These examples highlight cases where the captions are accurate, partially correct, or fail to capture the main scene, providing deeper insight into each model's strengths and weaknesses.

Example 1 Figure 4 shows a large crowd scene which includes a public demonstration in support of Palestine. The CRAFT caption (**"A photo of a demonstration in Beirut following the events of September 1970."**) uses the correct event term *demonstration* and gives a clear, precise description of the scene, and it also gives a year and location for the image. By contrast, the AraViT-GPT caption (**"An image of part of the Palestinian activities"**) is generic, does not explicitly describe the event, and reads less fluently. CRAFT provides a more accurate and informative caption for this image.

Example 2 Figure 5 shows a group of individuals in military uniforms gathered around the Palestinian leader Yasser Arafat while wearing sunglasses in a training camp. The CRAFT model produced the caption: **"An image of Yasser Arafat, Farouk Qaddoumi, and Ismail Shammout in one of the Palestinian revolution camps"**. This output demonstrates the model's ability to correctly identify Yasser Arafat, who is wearing sunglasses, and the training camp. However, it also introduces factual errors by naming two additional individuals (Farouk Qaddoumi and Ismail Shammout) who are not confirmed to be present in the image. The AraViT-GPT model captioned the image as: **"An image of a parade of Palestinian Liberation Army soldiers in one of the training camps"**, which is more generic, omits any individual identification, and focuses solely on the setting.

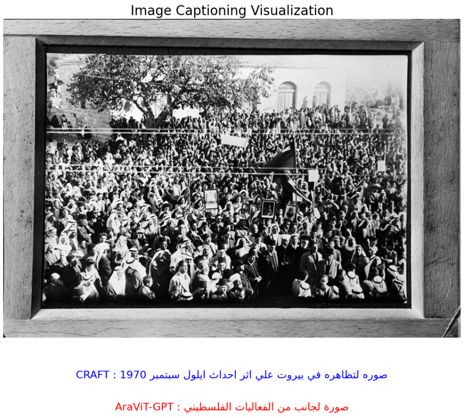


Image Captioning Visualization
CRAFT : صوره لتظاهره في بيروت علي اثر احداث ايلول سبتمبر 1970
AraViT-GPT : صورة لجانب من الفعاليات الفلسطيني

Figure 4: A photo of a historical demonstration in support of Palestine.

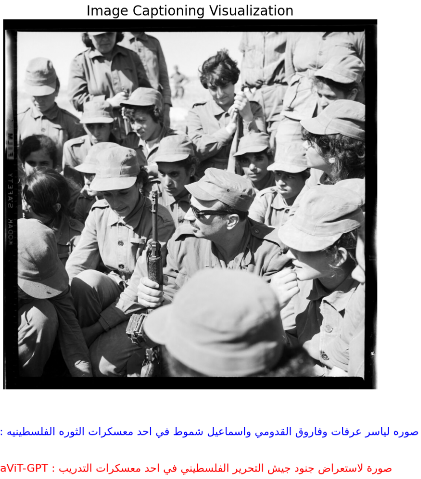


Image Captioning Visualization
CRAFT : صوره لياسر عرفات وفاروق القدومي واسماعيل شموط في احد معسكرات الثوره الفلسطينيه
AraViT-GPT : صورة لاستعراض جنود جيش التحرير الفلسطيني في احد معسكرات التدريب

Figure 5: A photo of a training camp involving members of the Palestinian revolution and Yasser Arafat in the middle of the group

6 Ablation study

As part of our ablation study, we implemented an intermediate image captioning system using a Vision-Encoder–Decoder architecture. This was not the final model reported in our main results, but it served to evaluate the performance trade-offs of combining a vision transformer encoder with a transformer-based autoregressive decoder.

Encoder The encoder component uses the google/vit-base-patch16-224 Vision Transformer (Dosovitskiy et al., 2021), which processes input images into a fixed-length sequence of visual embeddings. Images are preprocessed using ViTImageProcessor to ensure consistent scaling, normalization, and patch segmentation.

Decoder The decoder is initialized from a pretrained GPT-2 model (Radford et al., 2019). Since GPT-2 was originally trained with an English tokenizer, we replace its tokenizer with aubmindlab/bert-base-arabertv2 (Antoun et al., 2020) to enable high-quality Arabic caption generation. The decoder's embedding layer is resized to match the Arabic tokenizer's vocabulary size, and a new padding token is introduced to handle sequence batching.

Tokenizer Adaptation To accommodate GPT-2's architecture (Radford et al., 2019) with the Arabic tokenizer, the model configuration is updated to set decoder_start_token_id, eos_token_id, and pad_token_id appropriately. This ensures proper autoregressive decoding in Arabic.

Integration The encoder's output embeddings are passed to the decoder through the VisionEncoderDecoderModel framework from HuggingFace Transformers (Wolf et al., 2020). Training optimizes the cross-entropy loss over token predictions, ignoring padding tokens via masking.

7 Conclusion

In this paper, we developed CRAFT, a custom Arabic image captioning model that integrates Faster R-CNN region features, a custom vision transformer encoder, and a transformer decoder. The system was trained on a custom human-annotated dataset focused on the Palestinian narrative. We also developed AraViT-GPT, another Arabic captioning model, to evaluate against CRAFT. The evaluation results show that CRAFT excelled in semantic similarity, which was our primary metric, achieving a cosine similarity score of 56.22 on the test set. In contrast, AraViT-GPT achieved slightly higher BLEU and LLM judge scores.

We also demonstrated that CRAFT was able to identify people, locations, artifacts, and many other objects in the images. Despite the strengths of CRAFT, it has limitations, including occasional factual inaccuracies in named entities and limited lexical overlap with reference captions.

Future work will focus on scaling the dataset with more diverse Arabic captions, refining named entity recognition through multimodal pretraining, and incorporating nucleus sampling to improve caption fluency.

8 Limitations

While our approach achieved a good performance, several limitations remain. First, the dataset size is relatively small compared to standard benchmarks in image captioning, which restricts the generalization capacity of large-scale transformer models. Second, the captions are single reference annotations; it would be better for the model to have multiple references per image in order to capture the variability of natural language and allow fairer evaluation. Finally, our experiments were conducted on a single GPU (NVIDIA T4), which constrained the scale of hyperparameter exploration and limited the feasibility of training larger models.

9 Acknowledgments

We acknowledge the organizers of the shared task, the many contributors and annotators who enabled this work, and the anonymous reviewers whose comments improved the manuscript.

References

Moustafa Al-Hajj and Mustafa Jarrar. 2022. Arabglossbert: Fine-tuning bert on context-gloss pairs for wsd. *arXiv preprint arXiv:2205.09685*.

Rabee Al-Qasem, Mohannad Hendi, and Banan Tantour. 2025. Alkafi-llama: Fine-tuning llms for precise legal understanding in palestine. *Discover Artificial Intelligence*, 5(107).

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC*.

Ahlam Bashiti, Alaa Aljabari, Hadi Hamoud, Md. Rafiul Biswas, Bilal Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouani. 2025. ImageEval 2025: The First Arabic Image Captioning Shared Task. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, and et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Mohannad Mohammad Izzat Hendi and 1 others. 2023. *Accurate Pedestrian Detection for Human Crowds Using Deep Learning Techniques*. Ph.D. thesis, AAUP.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Sébastien Marcel and Yann Rodriguez. 2010. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, and 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*.

A Appendix

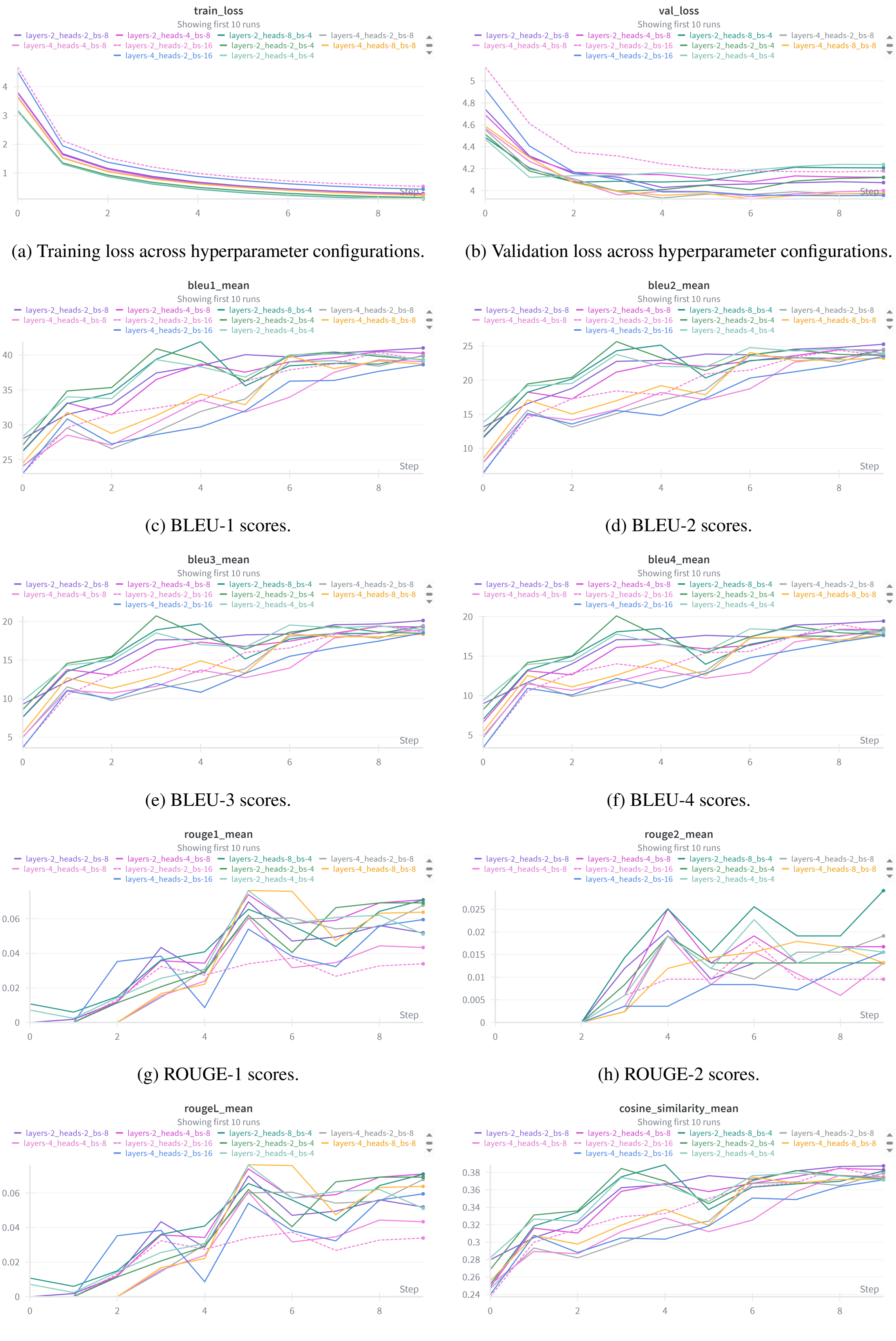
A.1 Final parameters used in training

Table 3: Training hyperparameters used in the experiments.

Parameter	**Value**
Number of epochs	40
Batch size	8
Encoder layers	2
Decoder layers	2
Attention heads (encoder)	2
Attention heads (decoder)	2
Learning rate	1×10^{-4}
Optimizer	AdamW
Learning rate schedule	Linear
Loss function	Cross Entropy Loss
Max sequence length	97
Input features	50 regions

A.2 Grid search tuning results

This section presents the results of our grid search experiments, showing how different hyperparameter configurations affected training and validation loss, as well as BLEU, ROUGE, and cosine similarity scores. The plots illustrate the trade-offs that guided our choice of the final configuration.



(a) Training loss across hyperparameter configurations.

(b) Validation loss across hyperparameter configurations.

(c) BLEU-1 scores.

(d) BLEU-2 scores.

(e) BLEU-3 scores.

(f) BLEU-4 scores.

(g) ROUGE-1 scores.

(h) ROUGE-2 scores.

(i) ROUGE-L scores.

(j) Cosine similarity across configurations.

Figure 6: Loss curves and evaluation metrics across hyperparameter configurations during the grid search.

VLCAP at ImageEval 2025 Shared Task: Multimodal Arabic Captioning with Interpretable Visual Concept Integration

Passant Elchafei
Ulm University, Germany
passant.elchafei@uni-ulm.de

Amany Fashwan
Alexandria University, Egypt
amany.fashwan@alexu.edu.eg

Abstract

We present VLCAP, an Arabic image captioning framework that integrates CLIP-based visual label retrieval with multimodal text generation. Rather than relying solely on end-to-end captioning, VLCAP grounds generation in interpretable Arabic visual concepts extracted with three multilingual encoders, mCLIP, AraCLIP, and Jina V4, each evaluated separately for label retrieval. A hybrid vocabulary is built from training captions and enriched with about 21K general domain labels translated from the Visual Genome dataset, covering objects, attributes, and scenes. The top-k retrieved labels are transformed into fluent Arabic prompts and passed along with the original image to vision–language models. In the second stage, we tested Qwen-VL and Gemini Pro Vision for caption generation, resulting in six encoder–decoder configurations. The results show that mCLIP + Gemini Pro Vision achieved the best BLEU-1 (5.34%) and cosine similarity (60.01%), while AraCLIP + Qwen-VL obtained the highest LLM-judge score (36.33%). This interpretable pipeline enables culturally coherent and contextually accurate Arabic captions.

1 Introduction

In today's digital age, images are everywhere, shared across social media platforms, embedded in news articles, used in education, e-commerce, and communication. With the rapid growth of visual content, images have become a dominant form of information exchange and expression. This widespread presence highlights the need for intelligent systems that can understand, interpret, and describe visual content effectively (Gendy and Patel, 2024).

Image captioning is the task of automatically generating syntactically correct and semantically meaningful sentences that describe an image's content. It plays a vital role in bridging the gap between visual content and natural language. Equipping machines with the ability to interpret and describe visual information offers numerous benefits, including enhanced information retrieval, support for early childhood education, assistance for visually impaired individuals, and applications in social media, among others. While understanding the content of an image may seem effortless, even for children, it remains a significant challenge for computers (Eljundi et al., 2020).

Vision-language (VL) models have significantly advanced image understanding and captioning tasks in English and other high-resource languages (Zhang et al., 2024). However, Arabic image captioning remains underexplored, particularly for culturally rich datasets requiring grounded and interpretable visual understanding. End-to-end generation models often fail to capture the fine-grained semantics, contextual nuances, and socio-cultural cues inherent in Arabic visual scenes.

This paper presents our submission to the ImageEval Shared Task, specifically the Image Captioning Models Evaluation Subtask. The objective of this subtask is to develop Arabic image captioning models capable of generating culturally relevant and contextually accurate image descriptions. To address this task, we propose VLCAP, a modular Arabic captioning pipeline that integrates visual label reasoning with multimodal generation. Unlike prior work, which mainly adapts English datasets or relies on end-to-end models, our approach explicitly grounds captioning in Arabic visual concepts. In the first stage, we conduct three separate experiments using CLIP-based encoders AraCLIP (Al-Barham et al., 2024), mCLIP (Chen et al., 2023), and Jina V4 (Günther et al., 2025) to extract the top-k Arabic visual labels for each image. These labels act as interpretable anchors representing "what is seen."

In the second stage, we use these extracted labels to construct enriched prompts, which are then

Proceedings of The Third Arabic Natural Language Processing Conference, pages 408–413
November 8-9, 2025 ©2025 Association for Computational Linguistics

combined with the original images and fed into two different vision–language models in separate experiments: Qwen-VL (Bai et al., 2023) and Gemini Pro Vision (Anil et al., 2023). This setup enables us to systematically evaluate which combination of label extraction model and caption generation model yields the most culturally aligned, semantically accurate, and fluent Arabic captions. By decoupling visual recognition from linguistic description, VLCAP improves both cultural relevance and model transparency.

The rest of this paper is organized as follows: Section 2 reviews the background and related work on Arabic image captioning and the dataset used in our study, while Section 3 presents VLCAP, our proposed Arabic vision–language captioning system, and describes the system overview. Section 4 discusses the results of our experiments. Finally, Section 5 concludes the paper.

2 Background

The ImageEval 2025 Shared Task focuses on evaluating image captioning models for the Arabic language, with two main subtasks: (1) Building an open-source dataset of images with culturally appropriate, naturally written Arabic captions, supporting the development of Arabic-native image captioning resources and (2) Automatic generation of Arabic captions for given images. We participated in Subtask 2: Image Captioning Models Evaluation, which requires participants to generate captions for a set of images in Arabic (Bashiti et al., 2025).

In Subtask 2, the input is a single image, and the output is a short, descriptive caption in ***Arabic***. Captions are submitted in CSV format, with each row containing the image_id and the generated caption. The generated captions are evaluated against a hidden reference set using a combination of automatic metrics ROUGE, BLEU and LLM-as-a-judge to assess semantic similarity and overall quality. The images cover a broad range of everyday scenes, enabling the models to learn both literal and contextually enriched descriptions. The dataset is culturally relevant, reflecting both Modern Standard Arabic and occasional dialectal expressions.

The dataset used in this shared task is entirely in Arabic and consists of high-quality, manually curated captions. It includes a training set of 2,718 images with human-authored captions, a validation set of 76 images with undisclosed gold-standard captions, and a test set of 753 images, also with undisclosed gold-standard captions.

Research on Arabic image captioning has been steadily growing, although it still lags behind progress in English captioning. Early work in the field primarily involved adapting English datasets by translating captions into Arabic or creating Arabic versions of existing datasets like Flickr8k and MS-COCO. (Al-muzaini et al., 2018) developed an Arabic image captioning dataset and implemented a model using a convolutional neural network (CNN) for image feature extraction and a recurrent neural network (RNN) with an LSTM decoder for sentence generation. These models laid the foundation for subsequent developments in Arabic captioning. (Eljundi et al., 2020) proposed an end-to-end model that directly transcribes images into Arabic text. They developed an annotated dataset for Arabic image captioning (AIC). They also developed a base model for AIC that relies on text translation from English image captions. The two models are evaluated with the new dataset, and the results show the superiority of their end-to-end model.

More recent studies have focused on building specialized datasets and improving model architectures. (Al-Malki and Al-Aama, 2023) built 'ArabicFashionData' dataset, which contains labeled images of clothing items. Using this data, researchers developed an attention-based encoder-decoder model that achieved a high BLEU-1 score of 88.52. (Za'ter and Talafha, 2022) highlighted the lack of standardized Arabic benchmarks and proposed unified datasets for evaluating multi-task learning approaches using pre-trained word embeddings, which showed moderate improvements in caption quality.

Transformer-based models have also gained traction. (Emami et al., 2022) explored Arabic image captioning using deep bidirectional transformers by integrating pre-trained language models into the generation process. (Alsayed et al., 2023) expanded on this by analyzing the impact of text preprocessing tools like CAMeL Tools and various image encoders such as ResNet152. Their experiments demonstrated substantial improvements in BLEU-4 scores up to 148% and their best-performing model outperformed existing approaches by 379%.

Vision-language models have introduced further advancements. The VIOLET model (Mohamed et al., 2023) combines a vision encoder with a Gemini text decoder. It leverages an automated

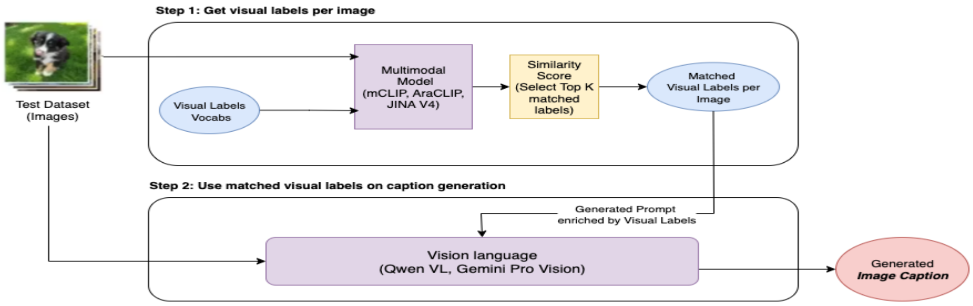

Step 1: Get visual labels per image
Test Dataset (Images)
Visual Labels Vocabs
Multimodal Model (mCLIP, AraCLIP, JINA V4)
Similarity Score (Select Top K matched labels)
Matched Visual Labels per Image
Step 2: Use matched visual labels on caption generation
Generated Prompt enriched by Visual Labels
Vision language (Qwen VL, Gemini Pro Vision)
Generated Image Caption

Figure 1: **VLCAP system overview.** The framework operates in two stages: (1) Arabic visual labels are retrieved by computing image–text similarity with a multilingual multimodal encoder (mCLIP, AraCLIP, or Jina V4) against a curated label vocabulary; (2) the retrieved labels are inserted into an Arabic prompt, which together with the original image, is passed to a vision–language model (Qwen-VL or Gemini Pro Vision) to generate the final caption.

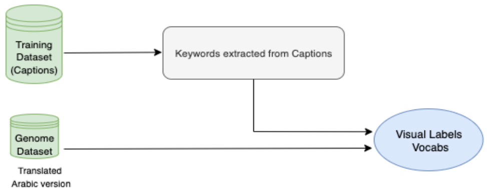

Training Dataset (Captions)
Keywords extracted from Captions
Genome Dataset
Translated Arabic version
Visual Labels Vocabs

Figure 2: **Visual Labels Vocab Builder**. Construction of the Arabic visual label vocabulary: Most frequent content words extracted from the training captions and augmented with general-domain visual concepts translated from the Visual Genome dataset, producing the final vocabulary used for label retrieval.

method to collect Arabic caption data from English sources, resulting in strong performance, including a CIDEr score of 61.2 on a manually annotated Arabic dataset.

In another recent approach, (Elbedwehy and Medhat, 2023) experimented with combining visual features extracted from powerful image encoders like SWIN, ConvNeXt, and XCIT, alongside Arabic pre-trained language models such as CAMeLBERT and MARBERTv2. This feature fusion strategy significantly improved the fluency and accuracy of the generated captions, outperforming earlier models.

3 System Overview

Unlike prior work that mainly adapts English datasets or relies on end-to-end models, our approach explicitly grounds captioning in Arabic visual concepts. We present **VLCAP**, a modular Arabic image captioning framework that decouples visual label extraction from caption generation to enhance cultural alignment, semantic accuracy, and interpretability. The system operates in two main stages as shown in Figure 1, supported by a comprehensive Arabic visual vocabulary constructed through the Visual Label Builder (Figure 2).

3.1 Arabic Visual Vocabulary Construction

As a preliminary step, we built a vocabulary of Arabic visual labels from two sources: **(1)** most frequent content words extracted from the training captions after removing Arabic stopwords, numbers, and very short tokens, and **(2)** an augmented set of 21,000 high-frequency visual concepts, covering objects, attributes, and scenes, translated from the Visual Genome dataset (Krishna et al., 2017) and adapted to Arabic cultural contexts. This vocabulary serves as the foundation for all subsequent label matching.

3.2 Visual Label Extraction

During inference (Figure 1), for each input image, multimodal similarity scores are computed between image embeddings and vocabulary label embeddings using three CLIP-based encoders: AraCLIP, mCLIP, and Jina V4. Cosine similarity ranking is applied to select the top-k matched labels per image, which serve as interpretable visual representations of “what is seen.” These matched labels are stored for later use in caption generation. Three separate experiments, one for each CLIP-based encoder, are conducted to determine the most effective label extractor.

Model	BLEU-1 Mean	Cosine Similarity Mean	LLM Judge Score
Base Model (Bashiti et al., 2025)	**16.98**	58.46	30.82
Gemini+mCLIP	5.34	**60.01**	33.05
Gemini+AraCLIP	4.25	58.89	**36.33**
Gemini+Jina V4	4.49	57.81	34.80
Qwen+mCLIP	5.20	58.39	23.49
Qwen+AraCLIP	4.57	57.19	31.40
Qwen+Jina V4	4.17	57.03	30.35

Table 1: Performance comparison of our CLIP-augmented captioning system (Gemini and Qwen combined with mCLIP, AraCLIP, and Jina V4) against the Base Model.

Participant	Cosine Similarity Mean
Base Model	58.46
VLCAP (**Ours**)	**60.01**
Averroes (Saeed et al., 2025)	58.55
phantom_troupe (Horaira et al., 2025)	57.48
ImpactAi (Al-Qasem and Hendi, 2025)	56.22
Codezone Research Group (Bichi et al., 2025)	38.30

Table 2: Cosine Similarity Mean scores for participating teams.

Participant	LLM Judge Score
Base Model	30.82
Averroes	**33.97**
VLCAP (**Ours**)	33.05
phantom_troupe	31.43
ImpactAi	26.55
Codezone Research Group	15.14

Table 3: LLM Judge Score results for participating teams.

3.3 Prompt-Guided Caption Generation

The top-ranked labels (typically 25–30) are used to construct an Arabic prompt of the form: “باستخدام العناصر التالية: *[top-ranked selected labels]* صف محتوى الصورة بدقة.”. This prompt, along with the input image, is fed into a vision–language model. We experiment with two models: Qwen-VL and Gemini Pro Vision. The resulting captions are grounded in both the visual content and the matched labels, ensuring cultural relevance and semantic accuracy.

3.4 Combination Analysis Evaluation

By pairing each of the three label extractors with both caption generation models, we evaluate a total of six configurations. The evaluation focuses on identifying the optimal combination for producing culturally aligned, semantically accurate, and fluent Arabic captions.

4 Results

The official evaluation of shared task submissions employed three complementary methods: Cosine Similarity, which quantifies lexical closeness between generated and reference captions after Arabic-specific normalization and TF–IDF n-gram comparison; LLM-as-a-Judge, using OpenAI’s GPT-4o to assess semantic accuracy, relevance, and fluency under reproducible conditions; and Manual Evaluation, where 5% of the test set was human-rated on cultural relevance, conciseness, completeness, and accuracy.

The results in Table 1 demonstrate that our system, which integrates CLIP-based visual label detection with Qwen and Gemini, consistently outperforms the base model (Bashiti et al., 2025) across all evaluation metrics. While the base model achieves the highest BLEU-1 score due to its direct captioning pipeline, it lags behind in semantic similarity and human-preference evaluations. For the Gemini model, mCLIP yields the strongest BLEU-1 and cosine similarity means, reflecting closer alignment with ground-truth captions and semantic coherence. Notably, AraCLIP, despite lower BLEU-1 and cosine similarity scores, achieves the highest LLM Judge Score, indicating that captions generated with AraCLIP labels are often judged as more contextually relevant or human-preferred. Jina V4 provides balanced performance across all metrics for Gemini. For Qwen, mCLIP again ranks highest in BLEU-1 and cosine similarity, but its

Participant	Cultural Relevance	Conciseness	Completeness	Accuracy
VLCAP	2.57	3.17	**2.67**	**2.97**
Averroes	**3.63**	**3.43**	2.60	2.80
Phantom Troupe	3.40	3.27	2.33	2.40
Codezone Research Group	1.10	2.03	1.47	2.03
ImpactAi	3.13	2.73	1.77	1.97

Table 4: Manual evaluation results based on Cultural Relevance, Conciseness, Completeness, and Accuracy.

LLM Judge Score is relatively low, suggesting that Qwen's outputs are less favored by human evaluators compared to Gemini. Conversely, AraCLIP and Jina V4 improve Qwen's LLM Judge Scores, highlighting the role of the CLIP-based label extractor in shaping user-perceived caption quality.

Overall, these results confirm that our CLIP-augmented system enhances performance beyond the baseline, particularly in semantic similarity and human preference, with the choice of CLIP model exerting a stronger influence on caption quality than the downstream vision–language model itself.

In this shared task, our system achieved the highest performance in the Cosine Similarity metric, ranking first among all participating teams (Table 2), and secured second place in the LLM-as-a-Judge evaluation (Table 3), reflecting strong results in both semantic adequacy and fluency. In the manual evaluation Table 4, our captions ranked first in Completeness (2.67%) and Accuracy (2.97%), while placing second in Cultural Relevance (2.57%) and Conciseness (3.17%). These outcomes underscore the system's ability to generate Arabic captions that are accurate, comprehensive, and semantically faithful, while maintaining competitive performance in cultural appropriateness and conciseness.

5 Conclusion

In this work, we introduced VLCAP, a modular Arabic image captioning framework that separates visual label extraction from caption generation to enhance cultural alignment, semantic accuracy, and interpretability. Through six experiments combining three CLIP-based encoders with two vision–language models, we found that Gemini Pro Vision + mCLIP delivered the strongest lexical and semantic performance (*BLEU-1:* 5.34, *cosine similarity:* 60.01), whereas Qwen-VL + AraCLIP achieved the highest LLM-based human-alignment score (36.33). These outcomes demonstrate that VLCAP's decoupled design allows tuning for different evaluation priorities and provides a transferable approach for culturally aware captioning in other low-resource languages.

References

Muhammad Al-Barham, Imad Afyouni, Khalid Almubarak, Ashraf Elnagar, Ayad Turky, and Ibrahim Hashem. 2024. Araclip: Cross-lingual learning for effective arabic image retrieval. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 102–110. Association for Computational Linguistics.

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

Huda Al-muzaini, Tasniem N., and Hafida Benhidour. 2018. Automatic arabic image captioning using rnn-lstm-based language model and cnn. *International Journal of Advanced Computer Science and Applications*, 9.

Rabee Al-Qasem and Mohannad Hendi. 2025. Impactai at imageeval 2025 shared task: Region-aware transformers for arabic image captioning: A case study on the palestinian narrative. In *Proceedings of the Third Arabic NLP Conference (ArabicNLP 2025)*. Co-located with EMNLP 2025, Nov 5–9.

Ashwaq Alsayed, Thamir M. Qadah, and Muhammad Arif. 2023. A performance analysis of transformer-based deep learning models for arabic image captioning. *J. King Saud Univ. Comput. Inf. Sci.*, 35(9).

Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and Orhan Firat. 2023. Gemini: A family of highly capable multimodal models.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond.

Ahlam Bashiti, Alaa Aljabari, Hadi Hamoud, Md. Rafiul Biswas, Bilal Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouani. 2025. ImageEval 2025: The First Arabic Image Captioning Shared Task. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*. Co-located with EMNLP 2025, November 5–9.

Abdulkadir Shehu Bichi, Ismail Dauda Abubakar, Fatima Muhammad Adam, Aminu Musa, Auwal Umar Ahmed, Abubakar Ibrahim, Khadija Salihu Aua, Aisha Mustapha Ahmed, and Mahmud Said Ahmed. 2025. Codezone research group at imageeval 2025 shared task: Arabic image captioning using blip and m2m100: A two-stage translation approach for imageeval 2025. In *Proceedings of the Third Arabic NLP Conference (ArabicNLP 2025)*. Co-located with EMNLP 2025, Nov 5–9.

Guanhua Chen, Lu Hou, Yun Chen, Wenliang Dai, Lifeng Shang, Xin Jiang, Qun Liu, Jia Pan, and Wenping Wang. 2023. mCLIP: Multilingual CLIP via cross-lingual transfer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13028–13043, Toronto, Canada. Association for Computational Linguistics.

Samar Elbedwehy and T. Medhat. 2023. Improved arabic image captioning model using feature concatenation with pre-trained word embedding. *Neural Comput. Appl.*, 35(26):19051–19067.

Obeida Eljundi, Mohamad Dhaybi, Kotaiba Mokadam, Hazem Hajj, and Daniel Asmar. 2020. Resources and end-to-end neural network models for arabic image captioning. pages 233–241.

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*, pages 40–51, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Wen Gendy and Dularia Patel. 2024. Advancements in computer vision: A comprehensive survey of image processing and interdisciplinary applications. *Academic Journal of Science and Technology*, 13(2):28–34.

Michael Günther, Saba Sturua, Mohammad Akram, Isabelle Mohr, Andrei Ungureanu, Sedigheh Eslami, Scott Martens, Bo Wang, Nan Wang, and Han Xiao. 2025. jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval.

Muhammad Abu Horaira, Farhan Amin, Sakibul Hasan, Md. Tanvir Ahammed Shawon, and Muhammad Ibrahim Khan. 2025. Phantomtroupe at imageeval shared task: Multimodal arabic image captioning through translation-based fine-tuning of llm models. In *Proceedings of the Third Arabic NLP Conference (ArabicNLP 2025)*. Co-located with EMNLP 2025, Nov 5–9.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David Shamma, Michael Bernstein, and Fei-Fei Li. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123.

Abdelrahman Mohamed, Fakhraddin Alwajih, El Moatez Billah Nagoudi, Alcides Alcoba, and Muhammad Abdul-Mageed. 2023. Violet: A vision-language model for arabic image captioning with gemini decoder. pages 1–11.

Mariam Saeed, Sarah Elshabrawy, Abdelrahman Hagrass, Mazen Yasser, and Ayman Khalafallah. 2025. Averroes at imageeval 2025 shared task: Advancing arabic image captioning with augmentation and two-stage generation. In *Proceedings of the Third Arabic NLP Conference (ArabicNLP 2025)*. Co-located with EMNLP 2025, Nov 5–9.

Muhy Eddin Za'ter and Bashar Talafha. 2022. Benchmarking and improving arabic automatic image captioning through the use of multi-task learning paradigm.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5625–5644.

PhantomTroupe at ImageEval 2025 Shared Task: Multimodal Arabic Image Captioning through Translation-Based Fine-Tuning of LLM Models

Muhammad Abu Horaira*, Farhan Amin*, Sakibul Hasan*,
Md. Tanvir Ahammed Shawon, Muhammad Ibrahim Khan
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2004029, u2004068, u2004043}@student.cuet.ac.bd

Abstract

Generating culturally accurate captions for images in Arabic remains a challenging task due to the language's rich morphology, complex syntax, and diverse cultural contexts.Cultural preservation involves capturing the significance and emotional resonance of images related to Palestinian heritage, ensuring accurate representation for future generations. We present a translation-assisted, instruction-tuned multimodal pipeline for Arabic image captioning, developed for the ImageEval 2025 Shared Task Subtask 2: on the evaluation of image captioning models.Our approach leverages the Qwen2.5-VL-7B-Instruct model with 4-bit quantization, fine-tuned using Parameter-Efficient Fine-Tuning (PEFT) with LoRA. We implemented a pipeline involving translation of Arabic captions to English, followed by back-translation to generate fluent Arabic outputs. We evaluated several vision-language models, including Qwen2.5 VL (7B), Llama 3.2 (11B), and Pixtral (12B). The Qwen2.5 VL (7B) model achieved a BLEU-1 score of 22.6, a Cosine Similarity of 57.48, and an LLM Judge Score of 31.43, securing third place in the competition. These results underscore the potential of instruction-tuned multimodal models to produce culturally sensitive Arabic captions.

1 Introduction

Image captioning, the task of generating natural language descriptions for visual content, has advanced rapidly with the rise of deep learning and vision-language models. While these techniques have achieved impressive results in English and other resource-rich languages, extending them to Arabic presents distinctive challenges due to the language's morphological richness, syntactic flexibility, and dialect diversity(Al-Khalifa et al., 2021). Arabic words often encode extensive grammatical information within a single token, variations in word order can influence fluency, and regional dialects differ both lexically and culturally. The ImageEval 2025, First Arabic Image Captioning Shared Task addresses these gaps by introducing the first open-source, manually captioned Arabic dataset, enabling the development of culturally relevant models (Bashiti et al., 2025). In this paper, we present the Phantom Troupe team's participation in Subtask-2,where we developed a translation-assisted, instruction-tuned multimodal pipeline using the Qwen2.5-VL (7B) model.

Our key contributions include:

- We Developed a bidirectional translation pipeline that significantly improved Arabic image captioning by leveraging multilingual pretraining data, producing more accurate and contextually relevant captions.

- We Analyzed the effects of preprocessing techniques (e.g., RGB versus grayscale inputs), translation quality, and Low-Rank Adaptation (LoRA) configurations to optimize model performance and understand their impact on caption quality.

The rest of the paper is organized as follows. Section 2 surveys related work on Arabic image captioning. Section 3 describes the dataset used in this study. Section 4 introduces our methodology, including preprocessing, translation, and fine-tuning steps. Section 5 details the parameter settings, while Section 6 presents results and error analysis. Section 7 discusses ethical considerations, Section 8 outlines limitations, and Section 9 concludes the paper.

2 Related Works

Arabic image captioning is a growing research area focused on creating natural language descriptions for images while respecting the unique features of the Arabic language. This involves dealing with its complex word forms, varied sentence structures, and regional dialects, as well as capturing cultural details to make the captions both accurate and meaningful. The challenges of Arabic captioning stem from its rich morphology, flexible syntax, and diverse dialects. Early advances in image captioning were shaped by the introduction of attention mechanisms (Xu et al., 2015), which have since been adapted for Arabic in multiple studies. A comprehensive review highlighted the need for culturally aware datasets and models tailored to Arabic's linguistic diversity (Al-Khalifa et al., 2021).

Several architectural innovations have been proposed to address these challenges. The *AraCap* framework combined convolutional and recurrent networks to improve fluency and semantic accuracy (Afyouni et al., 2021), while other approaches leveraged visual–textual feature concatenation with pretrained word embeddings for performance gains (Elbedwehy and Medhat, 2023). ResNet50-based visual backbones have also been explored for Arabic captioning tasks (Alazzam, 2022).

Training strategies have evolved alongside architectural improvements. Multi-task learning has been shown to boost caption quality (Za'ter and Talafha, 2022), and self-critical sequence training (SCST) (Rennie et al., 2017) has been adapted for Arabic contexts to refine generation through reinforcement learning. Transfer learning from large-scale vision–language models has further improved performance (Ibrahim et al., 2024), while comparative analyses have examined the impact of deep learning factors on accuracy and robustness (Hejazi and Shaalan, 2021). More recently, BLIP-based vision–language integration has demonstrated strong results for Arabic caption generation (Sayed et al., 2024).

Overall, existing work reflects steady progress in Arabic image captioning, yet also underscores the need for models that are not only computationally efficient but also linguistically and culturally efficient.

3 Dataset Description

We utilized the dataset provided for the Shared Task on Arabic Image Captioning with Cultural Relevance, part of ImageEval 2025 (Bashiti et al., 2025).The dataset contains 3,071 manually annotated images, with 2,717 used for training, 75 for validation and 279 reserved for testing. It includes manually written Arabic captions in the training set that capture the language and cultural details common in Arabic-speaking communities. The test set, provided without captions, ensures a blind evaluation process.Our generated captions were evaluated via the CodaLab platform using standard metrics such as BLEU, cosine similarity to assess their accuracy and cultural relevance.

4 Methodology

4.1 System

Our goal is to produce culturally accurate Arabic captions for historical and cultural images by preserving named entities, and details such as attire and artifacts. Figure 1 provides an overview of the full pipeline. It illustrates how the vision encoder, translation component, and fine-tuning process are linked together. This layout helps clarify how visual information flows into the model and is combined with the translated text before caption generation.

4.2 Image Preprocessing

We have converted all images to grayscale to maintain a consistent visual style and reduce computational complexity by removing non-essential color channels.While RGB inputs can offer richer visual information, our focus was on structural and textural features rather than color-based cues.As, our dataset contained relatively few RGB images, making grayscale a more uniform choice.All images were resized to 224×224 pixels to match the model's input requirements.

4.3 Translation

We translated the original Arabic captions into English using the unsloth/Qwen3-14B model and then back-translated the outputs into Arabic after fine-tuning. This loop improved clarity and fluency, as working in English helped the model generate more precise and descriptive captions while preserving meaning. We chose Qwen3-14B for its strong multilingual training, which made it more effective than alternatives like MarianMT, especially in retaining culturally significant expressions. Although we did not run a large-scale comparison, qualitative checks showed that Qwen3-14B reduced meaning loss during back-translation, which justified its inclusion in the pipeline.

4.4 Image-Caption Pairing

Each preprocessed image was paired with its translated English caption. We then used these image-caption pairs to fine-tune our model, ensuring that the training process learned to generate accurate

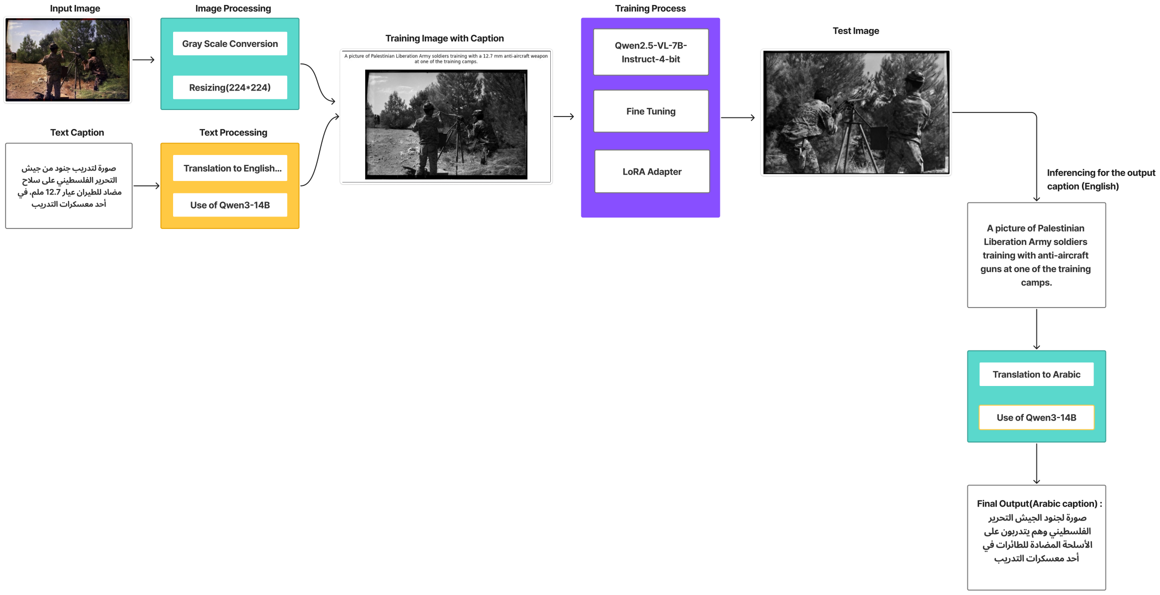
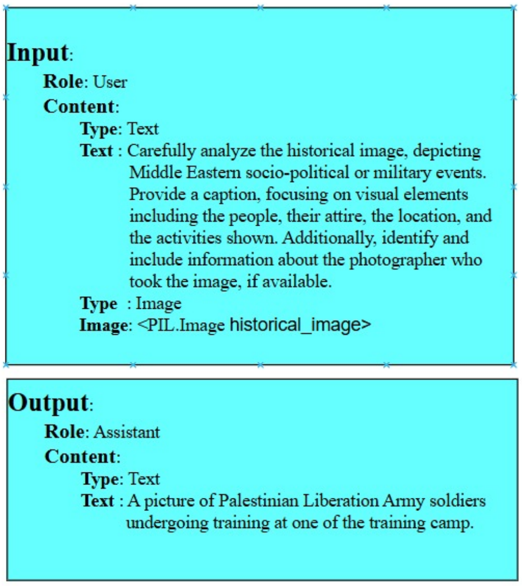

Input Image
Image Processing
Gray Scale Conversion
Resizing(224*224)
Text Caption
صورة لتدريب جنود من جيش
التحرير الفلسطيني على سلاح
مضاد للطيران عيار 12.7 ملم، في
أحد معسكرات التدريب
Text Processing
Translation to English...
Use of Qwen3-14B
Training Image with Caption
A picture of Palestinian Liberation Army soldiers training with a 12.7 mm anti-aircraft weapon at one of the training camps.
Training Process
Qwen2.5-VL-7B-Instruct-4-bit
Fine Tuning
LoRA Adapter
Test Image
Inferencing for the output caption (English)
A picture of Palestinian Liberation Army soldiers training with anti-aircraft guns at one of the training camps.
Translation to Arabic
Use of Qwen3-14B
Final Output(Arabic caption) :
صورة لجنود الجيش التحرير
الفلسطيني وهم يتدربون على
الأسلحة المضادة للطائرات في
أحد معسكرات التدريب

Figure 1: Multimodal architecture for Arabic image captioning using vision-language models



Input:

Role: User

Content:

Type: Text

Text : Carefully analyze the historical image, depicting Middle Eastern socio-political or military events. Provide a caption, focusing on visual elements including the people, their attire, the location, and the activities shown. Additionally, identify and include information about the photographer who took the image, if available.

Type : Image

Image: <PIL.Image historical_image>

Output:

Role: Assistant

Content:

Type: Text

Text : A picture of Palestinian Liberation Army soldiers undergoing training at one of the training camp.

Figure 2: Prompt provided to Qwen2.5-VL-7B-Instruct-bnb-4bit for caption generation

and culturally informed descriptions. Figure 2 illustrates instruction–response formatting for fine-tuning. This formatting ensured consistency between training and inference prompts for caption generation.

4.5 Initial Experimentation

We have evaluated several open-source multimodal and language models for Arabic image caption generation. Specifically,we experimented with unsloth/Llama-3.2-11B-Vision-Instruct and unsloth/Pixtral-12B-2409.Both models were loaded in 4-bit quantized format using the Unsloth library for memory efficiency and configured with gradient checkpointing to support longer context processing.Although these models produced syntactically valid Arabic captions, the outputs lacked semantic adequacy.Pixtral-12B demonstrated strong visual grounding, accurately capturing fine details, but had higher resource demands and slower training. Llama-3.2-11B-Vision converged faster but occasionally omitted culturally specific information BLEU and cosine similarity scores indicated suboptimal performance which motivated us to explore a model with stronger multilingual vision language alignment capabilities.

4.6 Overview of the Adopted Model

We adopted unsloth/Qwen2.5-VL-7B-Instruct-bnb-4bit as our final system due to its efficient multimodal integration, strong instruction-following capabilities, and relatively low computational cost compared to larger models. The model was loaded in 4-bit NF4 quantization using BitsAndBytes. LoRA adapters were applied to both vision and language heads with rank = 64, alpha = 64, and zero dropout, enabling parameter-efficient fine-tuning while keeping the base model largely frozen.

Fine-tuning was performed using the TRL `SFTTrainer` for 3 epochs with a batch size of 32, a learning rate of 5×10^{-5}, a cosine scheduler, and the AdamW 8-bit optimizer with weight decay of 0.01. Gradient checkpointing and FP16 mixed precision were used to reduce memory usage and accelerate training.

5 Parameter Setting

Initial Experiments:

LoRA rank = 16, alpha = 32, dropout = 0.05, batch size = 4, learning rate = 5×10^{-5}, 3 epochs.

Final Model

LoRA rank = 64, alpha = 64, dropout = 0.0, batch size = 32, learning rate = 5×10^{-5}, 3 epochs,

6 Results and Analysis

Table 1 compares the performance of the models on Arabic image captioning. We can see that Qwen 2.5-VL (7B) consistently outperforms both LLaMA 3.2 (11B) and Pixtral (12B). It achieves the highest BLEU-1 score (22.6), the best cosine similarity (57.48), and the highest LLM judge score (31.43), indicating that its captions are not only more accurate but also better aligned with human judgment.

Table 1: Evaluation Metrics for Arabic Image Captioning Models

Model	BLEU-1 Mean	Cosine Sim. Mean	LLM Judge Score
Qwen2.5 VL (7B)	22.6	57.48	31.43
Llama 3.2 (11B)	18.9	49.32	26.75
Pixtral (12B)	15.4	42.19	22.10

LLaMA 3.2 (11B) performs moderately well, but its captions sometimes miss finer cultural or contextual details. Pixtral (12B), struggles to generate semantically accurate and culturally relevant captions. Overall, these results highlight that Qwen 2.5-VL strikes the best balance between understanding the images and producing fluent, culturally aware Arabic captions, making it the most suitable choice for future enhancements in Arabic image captioning systems.

6.1 Error Analysis

Even though Qwen 2.5-VL generates high-quality captions, we noticed some recurring mistakes. Sometimes the model mixes up different Arabic dialects during translation, which can make the captions sound slightly inconsistent. It also occasionally drops culturally important terms. These errors show that while the model understands the images well, capturing the finer linguistic and cultural details in Arabic remains a challenge.

7 Ethical Considerations

For this study, we used the dataset provided in the shared task, which is publicly available. We ensured that all data usage complied with the task guidelines.Since Arabic is culturally and linguistically diverse, we paid special attention to avoid biased or offensive captions.We also recognize that automated captions may occasionally miss cultural or contextual nuances, so we recommend using them to support human judgment rather than replacing it, especially in sensitive contexts.

8 Limitations

Although the proposed pipeline performed well in the shared task, it has several limitations. Reliance on translation and back-translation makes the system dependent on intermediate translator quality, with errors sometimes propagating into final captions. Dialectal variation is also challenging, as the model often defaults to Modern Standard Arabic, limiting its reflection of regional varieties like Palestinian or Levantine Arabic. The dataset (3,071 images) is relatively small compared with large-scale English or Chinese resources, restricting generalization to unseen cultural contexts. Computational constraints also prevented testing larger models or diverse ensembles that could boost performance. Future work should explore larger, culturally diverse datasets, direct Arabic captioning, and improved handling of dialectal diversity.

9 Conclusion

This study has demonstrated the effectiveness of our approach to Arabic image captioning. By fine-tuning Qwen2.5-VL-7B-Instruct and employing translation-based training strategies, we achieved strong performance across multiple evaluation metrics. Integrating cultural preservation techniques and efficient fine-tuning proved essential for capturing the subtle linguistic details in Arabic captions.

Our comparison with other vision-language models highlights the clear advantage of instruction-tuned large language models in generating fluent, context-aware, and culturally sensitive descriptions. At the same time, challenges such as translation dependency and dialectal variations remain, pointing to opportunities for future work.

References

Imad Afyouni, Imtinan Azhar, and Ashraf Elnagar. 2021. Aracap: A hybrid deep learning architecture for arabic image captioning. *Procedia Computer Science*, 189:171–178.

Hend Al-Khalifa, Mustafa Jarrar, and Wajdi Zaghouani. 2021. Challenges in arabic image captioning: A review. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 20(4):1–25.

Batool Mohammed Alazzam. 2022. Arabic image captioning using resnet50. *Journal of Al-Qadisiyah for Computer Science and Mathematics*, 14(1):81–88.

Ahlam Bashiti, Alaa Aljabari, Hadi Hamoud, Md. Rafiul Biswas, Bilal Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouani. 2025. Imageeval 2025: The first arabic image captioning shared task. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Samar Elbedwehy and Tamer Medhat. 2023. Improved arabic image captioning model using feature concatenation with pre-trained word embedding. *Neural Computing and Applications*, 35(25):18575–18592.

Hani D. 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):37–44.

Haneen Siraj Ibrahim, Narjis Mezaal Shatia, and AbdulRahman A. Alsewari. 2024. A transfer learning approach for arabic image captions. *Mustansiriyah Journal of Science*, 35(1):70–79.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA. IEEE.

Abdelrahman M. Sayed, Mohamed K. Elhadad, Gouda I. Salama, and Aiman M. Mousa. 2024. Improving arabic image captioning with vision-language models. In *Proceedings of the 10th International Conference on Electrical Engineering and Informatics (ICEEI)*, Bandung, Indonesia. IEEE.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, Lille, France. PMLR.

Muhy Eddin Za'ter and Bashar Talafha. 2022. Benchmarking and improving arabic automatic image captioning through the use of multi-task learning paradigm. *arXiv preprint arXiv:2202.05474*.

NU_Internship team at ImageEval 2025: From Zero-Shot to Ensembles: Enhancing Grounded Arabic Image Captioning

Rana Gaber[1*] **Seif Eldin Amgad**[1*] **Ahmed Sherif Nasri**[2*]
Mohamed Ibrahim Ragab[3] **Ensaf H. Mohamed**[3]
[1] Faculty of Computers and Data Science, Alexandria University
[2] Faculty of Engineering, Ain Shams University
[3] CIS, School of Information Technology and Computer Science, Nile University
cds.{ranaahmed30309,seifamgad24237}@alexu.edu.eg
23p0270@eng.asu.edu.eg
{MoRagab, EnMohamed}@nu.edu.eg

Abstract

Arabic image captioning remains underexplored in vision–language research due to limited resources and the linguistic complexity of Arabic. In the ImageEval 2025 Shared Task, we evaluated three models, AIN, BLIP-Arabic-Flickr-8k, and Qwen 2.5, across zero-shot, fine-tuning, retrieval-augmented, and ensemble setups. Our official submission, fine-tuned BLIP with retrieval augmentation, ranked 5th overall based on both cosine similarity and LLM-as-a-judge scores. Post-submission experiments showed that ensemble captioning yielded the strongest captions across metrics. These findings demonstrate that even modest fine-tuning combined with retrieval augmentation can substantially improve Arabic captioning quality, which is significant in light of the limited resources for the language.

1 Introduction

Image captioning, the automatic generation of textual descriptions for visual content, has advanced significantly with the advent of large-scale vision–language models. While state-of-the-art systems achieve impressive performance in English and other high-resource languages, Arabic image captioning remains a challenging task due to its morphological complexity, limited annotated datasets, under-representation in multimodal benchmarks, lack of large-scale pretrained models, and tokenization compatibility issues. The difficulty is amplified in domain-specific contexts, where captions must reflect cultural, historical, and linguistic nuances accurately.

This study, conducted as part of ImageEval 2025 (Bashiti et al., 2025), focuses on a culturally and historically sensitive setting: generating Arabic captions for images related to the Palestinian Nakba. Producing accurate captions in this context requires not only linguistic fluency but also captions that remain faithful to historical narratives and avoid introducing misleading or invented details. Existing models often struggle with such specialized demands, leading us to assess which approaches perform best in this setting.

We present a comparative analysis of **AIN** (Heakl et al., 2025), **BLIP** (Li et al., 2022), and **Qwen 2.5 VL** (Bai et al., 2025) on the ImageEval 2025 Image Captioning Shared Task dataset. Each model is evaluated under four configurations: zero-shot with a RAG post-generation layer, fine-tuning, fine-tuning combined with RAG, and an LLM-based stacking ensemble for image captioning. The RAG component aims to improve domain relevance and factual grounding of the generated captions, while the stacking ensemble is designed to minimize errors by fusing captions produced by the top-performing models.

2 Background

2.1 Related Work

Prior work has explored transformer-based models, such as VIOLET (Mohamed et al., 2023), which employs a two-stage decoder for improved Arabic captioning. Additionally, multitask encoder–decoder approaches have been leveraged to enhance performance by leveraging action classification and pre-trained embeddings (Za'ter and Talafha, 2022).

2.2 Dataset

The dataset used in this study comprises 3,471 manually captioned images, primarily depicting events and scenes related to the Israeli–Palestinian conflict. It is divided into a training set of 2,718 images and a test set of 753 images. The training set was made available to participants for model development, while the test set was released later for automatic caption generation.

Each image is uniquely identified by its file-

[*] Equal contribution.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 419–431
November 8-9, 2025 ©2025 Association for Computational Linguistics

name (serving as its ID) and paired with a corresponding caption in a separate annotation file. The annotation file contains two columns—the textual description and the associated image filename—allowing direct mapping between captions and images.

The dataset is hosted on Hugging Face and distributed as part of the Image Captioning Shared Task 2025.

To expand our training data, we used Gemini-2.5-flash (Comanici et al., 2025) to paraphrase each caption twice, creating two additional captions per image. This allowed us to train with multi-reference captions. A custom Python script was developed to interact with the Gemini API, producing alternative Arabic captions that maintained the exact meaning of the originals. We tried various prompts on a small subset of the data, and the one that best preserved the original meaning was selected for generating the full dataset. This prompt, which ensured the quality and semantic consistency of the generated captions, is provided in Appendix A.1. Figure 1 shows an example image from the dataset with its human-written caption.

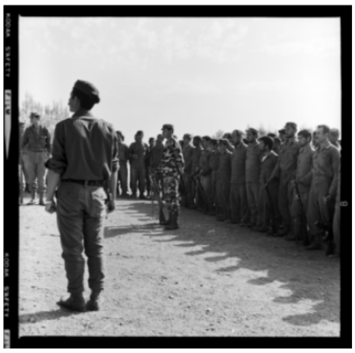


Figure 1: صوره لتدريب جنود جيش التحرير الفلسطيني في احد معسكرات التدريب.

3 System Overview

In our study, we conducted experiments utilizing three distinct models. The first is AIN (Arabic Inclusive Large Multimodal Model), which was developed by MBZUAI and trained on 3.6 million multimodal samples for English and Arabic captioning. The second is BLIP, a vision–language model extensively pre-trained on diverse web image–text datasets; we employed a publicly available variant from Hugging Face that had been fine-tuned on the Arabic Flickr8k dataset. The third is Qwen 2.5-VL, provided by the task organizers, which had already been fine-tuned and served as a benchmark for comparison in our analyses.

To enhance domain adaptability, we integrated a post-generation refinement layer inspired by (Ramos et al., 2023), adapting it to our task. A vector store was constructed from all Palestinian Nakba–related captions in the augmented training set, enabling the retrieval of examples with high semantic or lexical similarity to each generated caption. The retrieved examples, together with the original output, were provided to **Gemini-2.5-flash**, which revised the caption to align with the tone, style, and terminology of the retrieved material. The goal was to improve historical accuracy and stylistic consistency, while reducing obvious mistakes or hallucinations.

As the concluding phase in enhancing the quality of generated captions, we implemented an LLM-based stacking ensemble inspired by (Bianco et al., 2023). This approach involved providing captions generated by the models with the highest BLEU and cosine similarity scores to a meta-learner, utilizing the prompt detailed in Appendix A.3. This methodology facilitated the amalgamation of the most promising candidate captions, resulting in outputs that synthesized the strengths of multiple models while effectively mitigating their prevalent errors. The complete ensemble pipeline is illustrated in Figure 2.

We opted for the Gemini-2.5-flash model for both refinement and ensemble methodologies, owing to its generation quality. Additionally, its complementary tier rendered it a pragmatic choice for conducting iterative experimentation.

Building on the two methodologies described above, we designed four experiments aimed at systematically assessing and enhancing caption quality, namely:

- Zero-shot captioning with post-generation RAG
- Fine-tuned captioning
- Fine-tuned captioning with post-generation RAG
- LLM-based stacking ensemble

These approaches enabled a comparison of model performance across fine-tuning, retrieval-based contextual enhancement, combined methods, and ensemble learning.

Results reported for each configuration are post-submission results, obtained by submitting model outputs to the **CodaLab** evaluation server of the Image Captioning Shared Task 2025. Generated

GitHub Repository

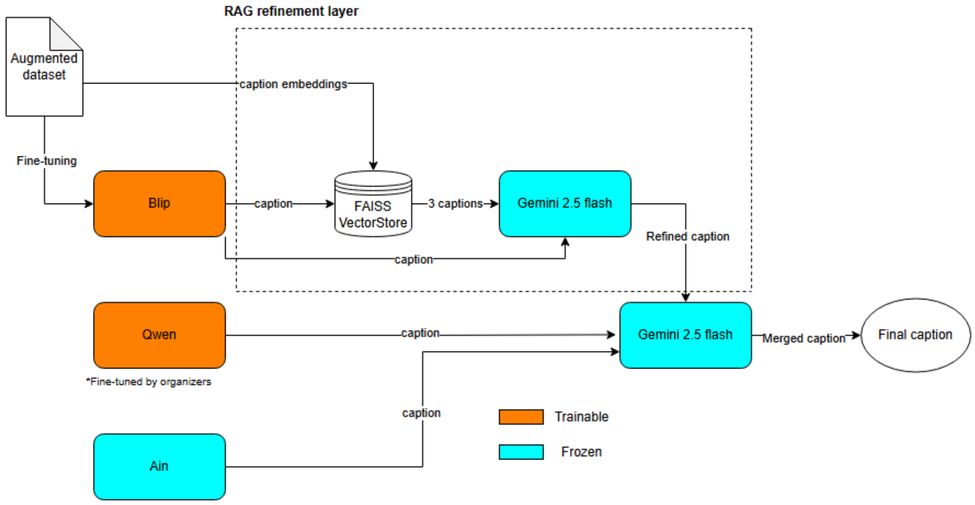


Figure 2: Pipeline diagram of the ensemble system. It integrates fine-tuned BLIP augmented with a RAG refinement layer, fine-tuned Qwen, and zero-shot Ain, with final caption fusion performed by Gemini.

captions were compared to ground-truth references using established metrics: BLEU (1–4) (Papineni et al., 2002), mean cosine similarity, and LLM-as-a-judge, (Wei et al., 2024), which, in this case, is **GPT-4o** (OpenAI et al., 2024). However, all official results are included in section 5.2.

4 Experiment Setup

All models under evaluation were fine-tuned using LORA (Hu et al., 2021) on the augmented Training dataset. All experiments were conducted using the Lightning.ai platform. For fine-tuning, AIN model was trained on NVIDIA L40S GPUs. while BLIP was trained on NVIDIA L4 GPUs. All training scripts were executed in a VS Code environment within Lightning.ai.

4.1 Data Preprocessing

Prior to training, all textual data was standardized using the Camel Tools library (Obeid et al., 2020) to ensure consistency and reduce orthographic variability in Arabic script. The preprocessing pipeline included Unicode and digit normalization, removal of diacritics, and orthographic unification (e.g., mapping آ, أ, إ to ا, ى to ي, and ة to ه). Elongation marks (Tatweel) were stripped, along with non-linguistic elements such as dates, numbers, and punctuation. Finally, whitespace was normalized by collapsing multiple spaces and trimming edges. These steps yielded clean, linguistically normalized input free of irrelevant tokens, leading to cleaner inputs and more dependable downstream results.

4.2 Model Fine-Tuning

The fine-tuning configuration is summarized in Table 1. **BLIP** was trained for three epochs with a batch size of 16 and a weight decay of 0.001. **AIN** was trained for ten epochs with a batch size of 64 and a higher weight decay to enhance its ability to generate high-quality Arabic captions. In contrast, **Qwen** was fine-tuned by the shared task organizers for fifteen epochs with a batch size of 16 using a cosine learning rate scheduler.

5 Results

5.1 Post-submission Results

This section reports the performance of the evaluated image captioning models under the four configurations described in Section 3, with zero-shot results included for comparison in Table 2. While traditional n-gram overlap metrics yielded relatively low scores, performance was higher on LLM-as-a-judge and cosine similarity, indicating that the generated captions were semantically related to the images but diverged from the ground truth in lexical choice.

Metric	BLIP	AIN-8B	Qwen-7B
BLEU-1 (mean)	3.58	3.5	**9.92**
BLEU-2 (mean)	1.57	1.17	3.23
BLEU-3 (mean)	0.95	0.64	1.90
BLEU-4 (mean)	0.78	0.44	1.33
Cosine Similarity (mean)	38.01	**59.69**	55.77
LLM-as-a-Judge	6.29	25.27	**27.11**

Table 2: Evaluation scores for zero-shot Model captioning

Model	Learning Rate	Batch Size	Epochs	Optimizer	Weight Decay	Loss Function
BLIP	2×10^{-4}	16	3	AdamW	0.001	Cross-Entropy
AIN	2×10^{-4}	64	10	AdamW	0.01	Cross-Entropy
Qwen 2.5	2×10^{-5}	16	15	AdamW	0	Cross-Entropy

Table 1: Fine-tuning hyperparameters for each evaluated model.

5.1.1 Zero-shot captioning with post-generation RAG

This experiment evaluated the effect of the RAG layer on zero-shot models without prior task-specific training. As shown in table 3, For BLIP, both cosine similarity and LLM-as-a-judge improved slightly, reaching **42.96** and **7.2**. For AIN and Qwen, only LLM-as-a-judge increased, with scores of **29.49** and **30.51**, while cosine similarity declined relative to the zero-shot baseline. Nonetheless, RAG improved BLEU_1 across all three models, with BLIP, AIN, and Qwen achieving **10.12**, **7.79**, and **10.28**, though the gain for Qwen was marginal. These findings suggest that RAG helped in some cases (like BLIP) but not in others, meaning the gains depend heavily on how each model integrates external context. Captioning examples are provided in Appendix B.

Metric	BLIP	AIN	Qwen-2.5 VL
BLEU-1 (mean)	10.12	7.79	**10.28**
BLEU-2 (mean)	3.73	3.25	4.42
BLEU-3 (mean)	2.31	2.0	2.75
BLEU-4 (mean)	1.86	1.41	1.89
Cosine Similarity (mean)	42.96	**55.15**	52.39
LLM-as-a-Judge	7.2	29.49	**30.51**

Table 3: Evaluation scores for zero-shot Model captioning with RAG.

5.1.2 Fine-Tuned model captioning

Fine-tuning improved model alignment with the Palestinian Nakba domain (Table 4), though the magnitude of improvement varied across models. BLIP demonstrated substantial gains, achieving a mean cosine similarity of **54.18** and an LLM-as-a-judge score of **22.99**. Qwen also improved, though less markedly, with a mean cosine similarity of **58.46** and an LLM-as-a-judge score of **30.82**. In contrast, AIN generalized poorly, reflecting weaker domain alignment, We suspect this may be because AIN was originally trained on broader multimodal data and struggled to adapt to the very specific Nakba-related captions, as both cosine similarity and LLM-as-a-judge scores declined. Captioning examples are documented in Appendix C.

Metric	BLIP	AIN	Qwen-2.5-VL
BLEU-1 (mean)	**21.40**	3.64	16.98
BLEU-2 (mean)	10.66	1.21	8.62
BLEU-3 (mean)	6.15	0.75	5.43
BLEU-4 (mean)	4.29	0.56	3.05
Cosine Similarity (mean)	54.18	52.92	**58.46**
LLM-as-a-Judge	22.99	15.66	**30.82**

Table 4: Evaluation scores for Fine-tuned Model captioning

5.1.3 Fine-tuned Captioning with Post-generation RAG

The integration of both domain adaptation techniques yielded a notable improvement in performance. As shown in table 5, BLIP showed only marginal gains over the raw fine-tuned model across n-gram overlap, cosine similarity, and LLM-as-a-judge scores, scoring **22.77**, **55.32** and **24.87** respectively. However, AIN exhibited a more nuanced increase in LLM-as-a-judge, accompanied by a slight decline in cosine similarity; however, its BLEU score increased significantly, rising to **8.25** compared to the raw fine-tuned model. By contrast, Qwen's performance declined slightly across both cosine similarity and LLM-as-a-judge metrics. Appendix D contains the captioning examples.

Metric	BLIP	AIN	Qwen-2.5-VL
BLEU-1 (mean)	**22.77**	8.25	14.23
BLEU-2 (mean)	11.29	3.2	7.44
BLEU-3 (mean)	6.34	2.02	5.09
BLEU-4 (mean)	4.39	1.41	3.63
Cosine Similarity (mean)	**55.32**	51.63	53.91
LLM-as-a-Judge	24.87	20.51	**26.52**

Table 5: Evaluation scores for Fine-tuned Model captioning with RAG

5.1.4 LLM-based stacking ensemble

To leverage complementary strengths across models, we fused captions generated by Zero-shot AIN, Fine-tuned Qwen, and Fine-tuned BLIP with RAG using the meta-learner described in Section 3. Although this ensemble did not achieve the highest BLEU score,only scoring a BLEU_1 score of **8.47**, it outperformed all non–zero-shot configurations in terms of semantic alignment and human-likeness, attaining the best cosine similarity **59.17** and LLM-as-a-judge **32.92** scores as shown in table 6. The captioning examples are presented in Appendix E.

Metric	Meta-Learner
BLEU-1 (mean)	8.47
BLEU-2 (mean)	4.08
BLEU-3 (mean)	2.29
BLEU-4 (mean)	1.5
Cosine Similarity (mean)	**59.17**
LLM-as-a-Judge	**32.92**

Table 6: Evaluation scores for LLM-based stacking ensemble (Meta-Learner)

Overall, the results highlight a clear performance hierarchy across the four approaches. RAG provided improvements in both zero-shot and fine-tuned settings, with its effect on zero-shot models being substantial, though still below the gains achieved through fine-tuning alone. When combined with fine-tuning, RAG yielded further gains across most models. Notably, while other models reached top performance in individual metrics, the ensemble consistently achieved near top results across most metrics, yielding the best overall performance on average.

5.2 Official Results

As mentioned in Section 1, our official results are based on the outputs of fine-tuned BLIP with RAG, which determined our ranking. The evaluation primarily relied on LLM-as-a-judge and cosine similarity metrics, yielding scores of **24.87** and **55.32**. Additionally, 5 percent of the test set was evaluated by humans using qualitative criteria, **cultural relevance**, **conciseness**, **completeness**, and **accuracy**, rated from 1 to 4, with definitions in Appendix F. Our model showed competitive performance, with conciseness achieving a score of **2.97** and cultural relevance **2.57**, while completeness and accuracy obtained scores of **2.13** and **2.23**, respectively, as shown in table 7.

Metric	Score
Cultural Relevance	2.57
Conciseness	2.97
Completeness	2.13
Accuracy	2.23

Table 7: Official human evaluation results for fine-tuned BLIP with RAG.

6 Conclusion

In conclusion, most reported results were obtained post-submission, whereas the official ranking relied exclusively on fine-tuned BLIP with a RAG layer, which achieved the highest BLEU score of **22.77**. The ensemble's meta-learner attained the top LLM-as-a-judge score of **32.92** and nearly matched zero-shot AIN in cosine similarity with **59.17**. The effect of RAG, however, varied across models: while it consistently acted as a refinement layer that enhanced outputs, its contribution was contingent on the strength of the underlying model.

7 Limitations

This study's limitations stem from computational and resource constraints. Conducted on the free tier of the Lightning.ai platform with only 15 GPU credits, our experiments were limited in scale and duration. This precluded exhaustive hyperparameter searches and constrained the number of training epochs for larger models. The post-generation RAG and ensemble layers, implemented with the rate-limited Gemini-2.5-flash API, required test set inferences to be batched across multiple days and reduced opportunities for extensive prompt engineering. Finally, while our LLM-based stacking ensemble achieved the best qualitative performance, its sequential inferences and reliance on an additional LLM meta-learner make it computationally expensive, resulting in high latency and memory demands. These factors limited its practicality for real-time, resource-constrained industrial deployment. In addition to these computational constraints, the ground-truth captions for the test data were hidden from participants, precluding the use of additional evaluation metrics that might have provided further insights into morphologically rich Arabic.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Ahlam Bashiti, Alaa Aljabari, Hadi Hamoud, Md. Rafiul Biswas, Bilal Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouani. 2025. ImageEval 2025: The

First Arabic Image Captioning Shared Task. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Simone Bianco, Luigi Celona, Marco Donzella, and Paolo Napoletano. 2023. Improving image captioning descriptiveness by ranking and llm-based fusion. *Preprint*, arXiv:2306.11593.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3290 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Ahmed Heakl, Sara Ghaboura, Omkar Thawkar, Fahad Shahbaz Khan, Hisham Cholakkal, Rao Muhammad Anwer, and Salman Khan. 2025. Ain: The arabic inclusive large multimodal model. *Preprint*, arXiv:2502.00094.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*.

Abdelrahman Mohamed, Fakhraddin Alwajih, El Moatez Billah Nagoudi, Alcides Inciarte, and Muhammad Abdul-Mageed. 2023. Violet: A vision-language model for Arabic image captioning with gemini decoder. In *Proceedings of ArabicNLP 2023*, pages 1–11, Singapore (Hybrid). 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.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

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.

Rita Ramos, Desmond Elliott, and Bruno Martins. 2023. Retrieval-augmented image captioning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3666–3681, Dubrovnik, Croatia. Association for Computational Linguistics.

Hui Wei, Shenghua He, Tian Xia, Andy Wong, Jingyang Lin, and Mei Han. 2024. Systematic evaluation of llm-as-a-judge in llm alignment tasks: Explainable metrics and diverse prompt templates. *ArXiv*, abs/2408.13006.

Muhy Eddin Za'ter and Bashar Talafha. 2022. Benchmarking and improving arabic automatic image captioning through the use of multi-task learning paradigm. *Preprint*, arXiv:2202.05474.

A Prompt Templates

A.1 Prompt to Expand Training Dataset

هذه هي التسمية الأصلية للصورة: «orig» من فضلك، اكتب
نوعين آخرين من التسمية باللغة العربية لوصف نفس الصورة،
بصياغة مختلفة ولكن المعنى ذاته.

A.2 Prompt used to Train our models

Analyze the content of this image, which is potentially related to the Palestinian Nakba and Israeli occupation of Palestine, and provide a concise and meaningful caption in Arabic — about 15 to 50 words. The caption should reflect the scene's content and emotional context, and should be natural and culturally appropriate. Do not include any English, metadata, or titles — the caption must be in Arabic.

A.3 Prompt sent to the Meta learner in the LLM-based stacking ensemble method

You are an expert in image captioning. Your task is to merge three captions (one descriptive zero-shot caption and two domain-specific captions) into a single final caption.

Rules:

- Preserve and prioritize important domain-specific terminology and keywords from the domain captions, since they are valuable for matching hidden reference captions.
- Use the zero-shot caption to add descriptive details and improve fluency, but keep the result concise and natural.

- Conflict resolution rule:
  - If the domain captions conflict with each other or with the zero-shot caption, treat the zero-shot caption as the reference and adjust the domain information accordingly.
  - If the conflict is minor (e.g., synonyms or phrasing), prefer the domain wording.
- Avoid redundancy, contradictions, or overlong sentences.
- Keep the result concise (one short sentence if possible) and avoid redundancy.
- Output one final caption that balances accuracy, fluency, and semantic richness.

Captions to merge:

- Zero-shot caption: zero
- Domain-specific caption 1: domain1
- Domain-specific caption 2: domain2

Output: return a single caption that integrates the strengths of all three inputs, maximizes semantic overlap and cosine similarity with likely hidden reference captions. caption:

B Zero-shot with RAG Captioning Examples

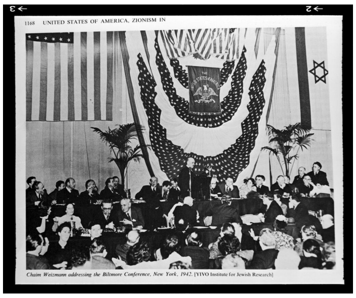

1168 UNITED STATES OF AMERICA, ZIONISM IN
Chaim Weizmann addressing the Biltmore Conference, New York, 1942. [YIVO Institute for Jewish Research]

Figure 3: Zero-shot with RAG on Example 1

AIN: الصورة تظهر مؤتمر بيلتمور في نيويورك عام 1942. يظهر فيها حاييم وايزمان يلقي خطابا امام عدد من الشخصيات الجالسة حول طاولات مستديرة. تبرز في الخلفية اعلام امريكية وعلم يحمل نجمة داود. الصورة من مكتبة ييفو للابحاث اليهودية.

Blip: صورة لمجموعة من الشباب الفلسطينيين يرتدون ملابس ملونة ويحملون الأعلام في مهرجان الشباب العربي بالجزائر

Qwen: تظهر الصورة اجتماعا في نيويورك عام 1942، حيث يتحدث الزعيم الصهيوني حاييم وايزمان عن القضية اليهودية وواقع الشعب الفلسطيني، وذلك في سياق المشروع الصهيوني. يظهر في الخلفية العلم الامريكي وعلم الحركة الصهيونية.

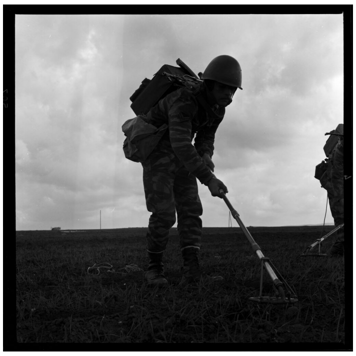


Figure 4: Zero-shot with RAG on Example 2

AIN: تظهر الصورة تدريب جنود من جيش التحرير الفلسطيني على سلاح مضاد للطيران عيار ملم في احد معسكرات التدريب.

Blip: صوره لرجل يحمل معدات عسكريه

Qwen: الصورة تظهر جندي الاحتلال وهو يقوم بتمشيط ارض خالية بحثا عن الالغام، ضمن جهود التعامل مع اثار الصراع في المنطقة

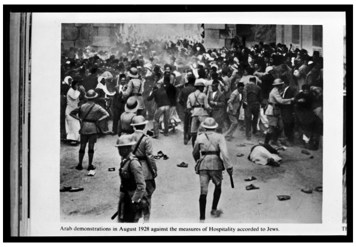

Arab demonstrations in August 1928 against the measures of Hospitality accorded to Jews.

Figure 5: Zero-shot with RAG on Example 3

AIN: صورة من عام 1928 لقمع قوات الأمن لمظاهرة عربية في المنطقة، احتجاجا على السياسات البريطانية الداعمة للاستيطان اليهودي. يظهر فيها رجال بزي عسكري مسلحين يواجهون المتظاهرين، حيث أسفرت المواجهات العنيفة عن سقوط أشخاص على الأرض.

Blip: سكان رفح يحيون الدبابات المصرية عند وصولها معبر رفح خلال حرب

Qwen: مظاهرات عربية في اغسطس 1928 احتجاجا علي سياسات الاحتلال البريطاني التي استهدفت حقوق ومقدسات المسلمين، وسط تواجد مكثف للقوات العسكرية التي قمعت المظاهرات.

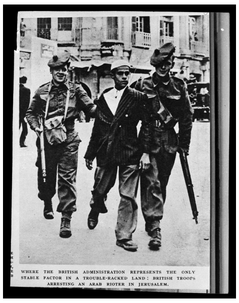

WHERE THE BRITISH ADMINISTRATION REPRESENTS THE ONLY STABLE FACTOR IN A TROUBLE-RACKED LAND: BRITISH TROOPS ARRESTING AN ARAB RIOTER IN JERUSALEM.

Figure 6: Zero-shot with RAG on Example 4

AIN: تظهر الصورة مشهدا من مدينة القدس عام 1948، حيث يقوم جنود الاحتلال البريطاني باعتقال متظاهر عربي. يظهر في الصورة ثلاثة جنود بريطانيين يرتدون الزي العسكري ويحملون البنادق، بينما يقتاد الرجل العربي الذي يرتدي ملابس تقليدية. يعكس هذا المشهد قمع الاحتلال البريطاني للمقدسيين خلال فترة الانتداب.

Blip: صوره لجنود الجيش المصري يسيرون في الشارع

Qwen: تظهر الصورة اعتقال جنود بريطانيين لرجل عراقي في القدس، وذلك خلال اضطرابات.

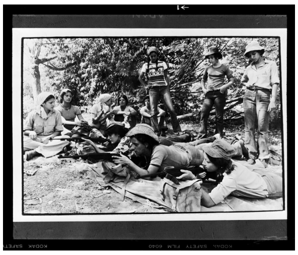


Figure 7: Zero-shot with RAG on Example 5

AIN: تظهر الصورة تدريب مجموعة من المقاتلين، بينهم نساء، على استخدام الأسلحة في أحد
معسكرات الثورة الفلسطينية، حيث يتخذون وضعيات تدريب مختلفة على الأرض.

Blip: صوره لناس يستلقون على الارض

Qwen: تظهر الصورة تجمعا لمجموعة من الاشخاص يجلسون علي الارض. تاريخ الصورة
ومصورها مجهولان.

C Fine-tuned Captioning Examples

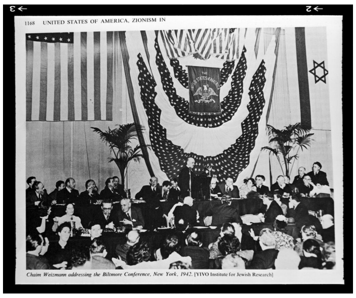

1168 UNITED STATES OF AMERICA, ZIONISM IN
Chaim Weizmann addressing the Biltmore Conference, New York, 1942. [YIVO Institute for Jewish Research]

Figure 8: Fine-tuned models on Example 1

AIN: تلتقط الصورة لحظة تاريخية في قاعة احتفالية في فلسطين عام 1942. في المقدمة، يجلس
الحضور في ملابس رسمية في مدرج، يراقبون الحدث الذي يجري في المسرح. مركز الصورة هو
الحدث الرئيسي الذي يجري في المسرح.

Blip: صورة لاحتفال عسكري لقوات الثورة الفلسطينية في بيروت

Qwen: صورة لاجتماع في نيويورك عام 1942

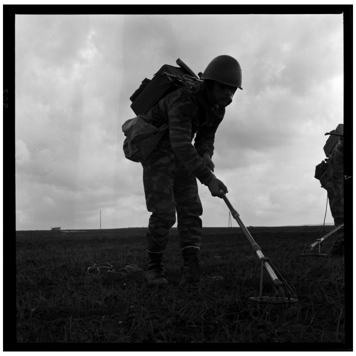


Figure 9: Fine-tuned models on Example 2

AIN: تلتقط الصورة مشهدا في منطقة عسكرية في منطقة القوقاز في عام 1992 خلال حرب
الشيشان. في المقدمة، يظهر جندي يرتدي زي عسكري ويحمل مسدس مثبت في حزامه. يحمل
جهازا يشبه جهاز الاستشعار في يديه، والذي يحتمل انه يستخدم في عمليات البحث عن المتفجرات.

Blip: صورة لتدريب مقاتلين الثورة الفلسطينية في احد معسكرات التدريب

Qwen: الصورة تظهر جنديًا يبحث عن الألغام في منطقة مستهدفة من قبل الاحتلال الإسرائيلي
خلال حرب حزيران/ يونيو 1967.

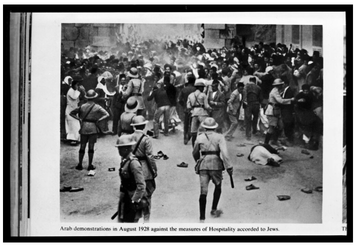

Arab demonstrations in August 1928 against the measures of Hospitality accorded to Jews.

Figure 10: Fine-tuned models on Example 3

AIN: تلتقط الصورة مشاهد مروعة من احتجاجات عربية في أغسطس عام 1928، احتجاجا
على التدابير البريطانية التي تدعم الاستيطان اليهودي في المنطقة. في المقدمة، يظهر جندي بريطاني
يركل متظاهرا عربيا بينما ترقد امرأة على الأرض في حالة صعبة.

Blip: صورة لاحتفال عسكري لقوات الثورة الفلسطينية في بيروت

Qwen: الصورة تظهر احتجاجًا عسكريًا من قبل السلطات الاستعمارية

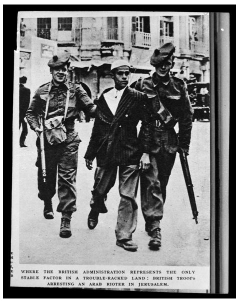

WHERE THE BRITISH ADMINISTRATION REPRESENTS THE ONLY STABLE FACTOR IN A TROUBLE-RACKED LAND: BRITISH TROOPS ARRESTING AN ARAB RIOTER IN JERUSALEM.

Figure 11: Fine-tuned models on Example 4

AIN: تلتقط الصورة مشهدا مثيرا في القدس عام 1948. في المقدمة، يقوم جنديان بريطانيان
باحتجاز رجل عربي يرتدي ملابس تقليدية، أحدهما يحمل بندقية والآخر مسدسا. يظهر في الخلفية
مشاة من المارة.

Blip: صورة لاحتفال عسكري لقوات الثورة الفلسطينية في بيروت

Qwen: الصورة تظهر جنديين بريطانيين يعتقمان شخصًا في القدس

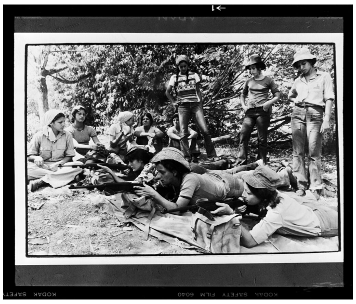


Figure 12: Fine-tuned models on Example 5

AIN: تلتقط الصورة مشهدا في الغابة حيث تشارك مجموعة من النساء في تدريبات عسكرية
تكتيكية، يتخذن وضعيات مختلفة للرماية.

Blip: صورة لاحتفال عسكري لقوات الثورة الفلسطينية في بيروت

Qwen: الصورة تظهر مجموعة من الفتيات اللاتي يتدربن على إطلاق النار، ويظهر في الصورة 10
فتاة، حيث يظهر في الصورة 5 فتيات يحملن البنادق ويستعدن لإطلاق النار، بينما يجلس الباقي في
الخلفية، ويظهر في الصورة شجرة خلف الفتيات.

Image	AIN	Blip	Qwen
Example 1	A picture of the Biltmore Conference in New York in 1942. Chaim Weizmann appears giving a speech to a number of figures sitting at round tables, with American flags and a flag with the Star of David in the background. The picture is from the YIVO Institute for Jewish Research.	A picture of a group of Palestinian youth in colorful clothes carrying flags at the Arab Youth Festival in Algeria.	A picture showing a meeting in New York in 1942, where the Zionist leader Chaim Weizmann speaks about the Jewish cause and the reality of the Palestinian people within the Zionist project, with the American flag and the Zionist movement's flag appearing in the background.
Example 2	A picture of Palestine Liberation Army soldiers training on an anti-aircraft weapon of a certain caliber in a training camp.	A picture of a man carrying military equipment.	A picture showing an occupation soldier combing an empty field in search of mines, within efforts to deal with the effects of conflict in the region.
Example 3	A picture from 1928 showing security forces suppressing an Arab demonstration in the region, protesting British policies supporting Jewish settlement. Armed men in military uniforms appear confronting demonstrators, where violent clashes occurred leading to the fall of individuals to the ground.	Rafah residents receiving Egyptian tanks entering Rafah crossing during the war.	Arab demonstrations in August 1928 rejecting the policies of the British occupation that targeted the rights and sanctities of Muslims, amid an intense military presence that suppressed the protests.
Example 4	A picture showing a scene from Jerusalem in 1948, where British occupation soldiers arrest an Arab protester. Three British soldiers in military uniforms carrying rifles appear, escorting an Arab man in traditional clothes, reflecting the suppression of Jerusalemites by the British during the Mandate period.	A picture of Egyptian army soldiers walking in the street.	A picture showing British soldiers arresting an Iraqi man in Jerusalem during the events.
Example 5	The photo shows a group of fighters, including women, being trained to use weapons in a Palestinian revolution camp, where they take different training positions on the ground.	A picture of people lying on the ground.	A picture showing a gathering of people sitting on the ground. The date and the photographer are unknown.

Table 8: English translations of captions generated by the models under the zero-shot with RAG configuration.

D Fine-tuned with RAG Captioning Examples

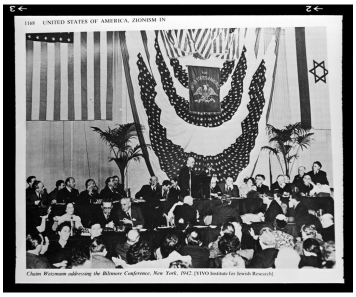

3←
2←
1168 UNITED STATES OF AMERICA, ZIONISM IN
THE UNITED STATES of AMERICA
Chaim Weizmann addressing the Biltmore Conference, New York, 1942. [YIVO Institute for Jewish Research]

Figure 13: Fine-tuned models with RAG on Example 1

AIN: تظهر الصورة جانبا من حدث تاريخي أقيم في قاعة احتفالات بفلسطين عام 1942. يجلس في المقدمة عدد من الحضور بملابس رسمية في مدرج، يتابعون الحدث الرئيسي الذي يعرض على المسرح. تبرز شخصية بارزة بين الحضور في الصف الأمامي.

Blip: صوره من احتفال عسكري لقوات الثوره الفلسطينيه في بيروت

Qwen: صورة توثق اجتماعاً في نيويورك عام 1942.

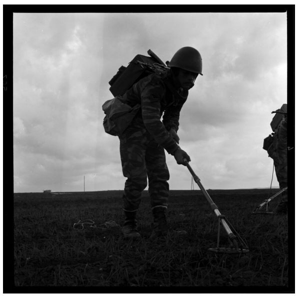


Figure 14: Fine-tuned models with RAG on Example 2

AIN: صورة لمشهد في منطقة عسكرية بالقوقاز عام 1992، تظهر جنديا يرتدي زيا عسكريا ويحمل مسدسا مثبتا في حزامه، كما يحمل جهازا يشبه جهاز استشعار يعتقد أنه يستخدم في عمليات البحث عن المتفجرات.

Blip: صوره لتدريب جنود جيش التحرير الفلسطيني في احد معسكرات التدريب

Qwen: صورة لجندي من قوات الاحتلال الإسرائيلي يبحث عن الألغام في منطقة خلال حرب حزيران/ يونيو 1967.

Image	AIN	Blip	Qwen
Example 1	The picture captures a historical moment in a ceremonial hall in Palestine in 1942. In the foreground, the attendees sit in formal clothing in a tiered seating, observing the event taking place on the stage. The center of the picture is the main event taking place on the stage.	A picture of a military celebration of the Palestinian revolution forces in Beirut.	A picture of a meeting in New York in 1942.
Example 2	The picture captures a scene in a military area in the Caucasus region in 1992 during the Chechen war. In the foreground, a soldier appears wearing a military uniform and carrying a pistol fixed in his belt. He is holding a device resembling a sensor in his hands, which is likely used in operations for detecting explosives.	A picture of the training of the Palestinian revolution fighters in one of the training camps.	The picture shows a soldier searching for mines in an area targeted by the Israeli occupation during the June 1967 war.
Example 3	The picture captures horrific scenes from Arab protests in August 1928, protesting the British measures supporting Jewish settlement in the region. In the foreground, a British soldier is kicking an Arab protester while a woman lies on the ground in a difficult state.	A picture of a military celebration of the Palestinian revolution forces in Beirut.	The picture shows a military protest by the colonial authorities.
Example 4	The picture captures an exciting scene in Jerusalem in 1948. In the foreground, two British soldiers are detaining an Arab man wearing traditional clothing, one of them holding a rifle and the other a pistol. In the background, pedestrians appear.	A picture of a military celebration of the Palestinian revolution forces in Beirut.	The picture shows two British soldiers arresting a person in Jerusalem.
Example 5	The picture captures a scene in the forest where a group of women are participating in tactical military training, taking different shooting positions.	A picture of a military celebration of the Palestinian revolution forces in Beirut.	The picture shows a group of girls who are training in shooting, and the picture shows 10 girls, where in the picture 5 girls carry rifles and are preparing to shoot, while the rest are sitting in the background, and in the picture there is a tree behind the girls.

Table 9: English translations of captions generated by the models under the fine-tuning configuration.

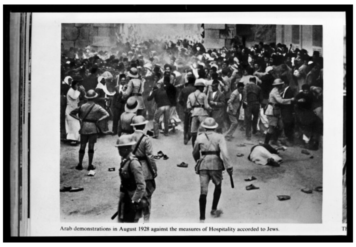

Arab demonstrations in August 1928 against the measures of Hospitality accorded to Jews.

Figure 15: Fine-tuned models with RAG on Example 3

AIN: تلتقط الصورة مشاهد مروعة من احتجاجات عربية في أغسطس عام 1928، ضد سياسات الانتداب البريطاني في المنطقة. يظهر جندي بريطاني وهو يركل متظاهرا في الشارع، بينما ترقد امرأة على الأرض في حالة إعياء.

Blip: صوره من احتفال عسكري لقوات الثوره الفلسطينيه في بيروت

Qwen: تُظهر الصورة احتجاجًا عسكريًا تنفذه السلطات الاستعمارية الصهيونية.

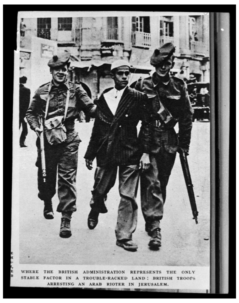

WHERE THE BRITISH ADMINISTRATION REPRESENTS THE ONLY STABLE FACTOR IN A TROUBLE-RACKED LAND: BRITISH TROOPS ARRESTING AN ARAB RIOTER IN JERUSALEM.

Figure 16: Fine-tuned models with RAG on Example 4

AIN: تظهر الصورة مشهدا من القدس عام 1948، خلال فترة الانتداب البريطاني. في المقدمة، يظهر جنديان بريطانيان بالزي الرسمي يقومان باحتجاز رجل وعربة، أحدهما يحمل بندقية والآخر مسدسا.

Blip: صوره من احتفال عسكري لقوات الثوره الفلسطينيه في بيروت

Qwen: صورة تُظهر جنديين بريطانيين يعتقلان شخصًا في القدس أثناء حرب.

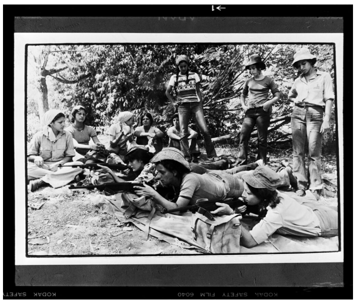


Figure 17: Fine-tuned models with RAG on Example 5

AIN: تظهر الصورة تدريب مقاتلات شابات ومقاتلين من قوات الثورة الفلسطينية على الرماية والتدريبات التكتيكية في أحد معسكرات التدريب بلبنان.

Blip: صوره من احتفال عسكري لقوات الثوره الفلسطينيه في بيروت

Qwen: صورة لمجموعة من عشر فتيات يتدربن على إطلاق النار، حيث تحمل خمس منهن بنادق ويستعددن للإطلاق، بينما تجلس الأخريات في الخلفية، وتظهر شجرة خلف الفتيات.

E LLM-based stacking Ensemble Captioning Examples

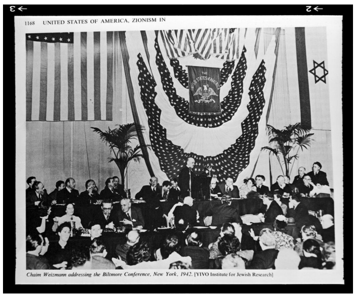

1168 UNITED STATES OF AMERICA. ZIONISM IN
Chaim Weizmann addressing the Biltmore Conference, New York, 1942. [YIVO Institute for Jewish Research]

Figure 18: LLM-based stacking Ensemble on Example 1

Meta Learner: صورة بالابيض والاسود لمؤتمر بيلتمور في نيويورك عام 1942، يظهر فيها شيمون وايزمان يلقي خطاباً أمام مجموعة من الرجال، مع وجود علم أمريكي كبير وعلم يحمل نجمة داود في الخلفية.

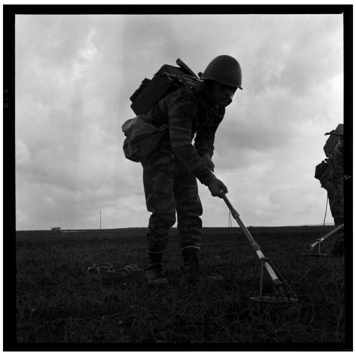


Figure 19: LLM-based stacking Ensemble on Example 2

Meta Learner: تظهر الصورة جنديين من جيش التحرير الفلسطيني يرتديان ملابس تمويهية، أحدهما يحمل جهازا يشبه المجرفة ويبحث عن الألغام في حقل عشبي ضمن منطقة مستهدفة من الاحتلال الإسرائيلي خلال حرب حزيران/ يونيو 1967.

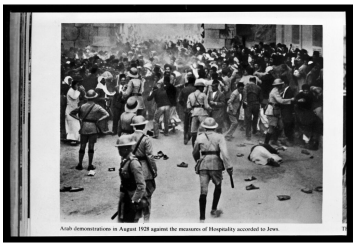

Arab demonstrations in August 1928 against the measures of Hospitality accorded to Jews.

Figure 20: LLM-based stacking Ensemble on Example 3

Meta Learner: تُظهر الصورة احتجاجًا عسكريًا متوترًا من عام 1928، حيث تواجه السلطات الاستعمارية المتظاهرين العرب اعتراضًا على إجراءات الضيافة الممنوحة للمستوطنين اليهود، مع وجود رجال يرتدون زي الجيش ويحملون أسلحة وأشخاص ملقون على الأرض.

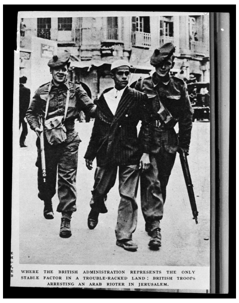

WHERE THE BRITISH ADMINISTRATION REPRESENTS THE ONLY STABLE FACTOR IN A TROUBLE-RACKED LAND: BRITISH TROOPS ARRESTING AN ARAB RIOTER IN JERUSALEM.

Figure 21: LLM-based stacking Ensemble on Example 4

Meta Learner: تظهر الصورة جنوداً بريطانيين يرتدون زياً عسكرياً ويعتقلون متظاهراً عربياً يرتدي ملابس تقليدية في القدس عام 1948.

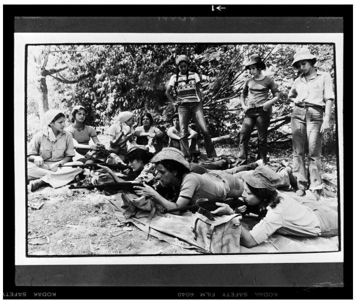


Figure 22: LLM-based stacking Ensemble on Example 5

Meta Learner: تظهر الصورة مجموعة من الفتيات، يرتدين ملابس غير رسمية، يشاركن في تدريب عسكري على إطلاق النار لقوات الثورة الفلسطينية في بيروت، حيث تحمل خمس منهن بنادق ويستعدن لإطلاق النار بينما تجلس البقية في الخلفية، مع شجرة خلفهن.

Image	AIN	Blip	Qwen
Example 1	The picture shows a side of a historical event held in a ceremonial hall in Palestine in 1942. In the foreground, several attendees in formal clothing sit in a tiered seating, following the main event presented on the stage. A prominent figure stands out among the attendees in the front row.	A picture from a military celebration of the Palestinian revolution forces in Beirut.	A picture documenting a meeting in New York in 1942.
Example 2	A picture of a scene in a military area in the Caucasus in 1992, showing a soldier wearing a military uniform and carrying a pistol fixed to his belt, also carrying a device resembling a sensor believed to be used in operations for detecting explosives.	A picture of the training of soldiers of the Palestinian Liberation Army in one of the training camps.	A picture of a soldier from the Israeli occupation forces searching for mines in an area during the June 1967 war.
Example 3	The picture captures horrific scenes from Arab protests in August 1928, against the policies of the British mandate in the region. A British soldier is shown kicking a protester in the street, while a woman lies on the ground in a state of exhaustion.	A picture from a military celebration of the Palestinian revolution forces in Beirut.	The picture shows a military protest carried out by the Zionist colonial authorities.
Example 4	The picture shows a scene from Jerusalem in 1948, during the British mandate. In the foreground, two British soldiers in official uniform are shown detaining a man and a cart, one of them holding a rifle and the other a pistol.	A picture from a military celebration of the Palestinian revolution forces in Beirut.	A picture showing two British soldiers arresting a person in Jerusalem during a war.
Example 5	The picture shows training of young female fighters and fighters from the Palestinian revolution forces in shooting and tactical training in one of the training camps in Lebanon.	A picture from a military celebration of the Palestinian revolution forces in Beirut.	A picture of a group of ten girls training on shooting, where five of them are carrying rifles and preparing to shoot, while the others sit in the background, and a tree appears behind the girls.

Table 10: English translations of captions generated by the models under the fine-tuning with RAG configuration.

F Human Assessment metric Definitions

Cultural Relevance – Measures whether the description reflects cultural specificity and provides contextual information related to the scene.

Conciseness – Assesses whether the description conveys information directly and succinctly, without unnecessary repetition or dispersion of details.

Completeness – Evaluates the extent to which the description covers all aspects of the image, including events, entities, and relevant elements.

Accuracy – Measures whether the description contains correct information, free from factual or conceptual errors.

Image	Meta Learner
Example 1	A black-and-white picture of the Biltmore Conference in New York in 1942, in which Chaim Weizmann appears giving a speech in front of a group of men, with a large American flag and a flag with the Star of David in the background.
Example 2	The picture shows two soldiers from the Palestinian Liberation Army wearing camouflage clothing, one of them holding a device resembling a shovel and searching for mines in a grassy field within an area targeted by the Israeli occupation during the June 1967 war.
Example 3	The picture shows a tense military protest from the year 1928, where the colonial authorities confronted the Arab demonstrators objecting to the hospitality measures granted to the Jewish settlers, with men wearing military uniforms carrying weapons and people lying on the ground.
Example 4	The picture shows British soldiers wearing military uniforms arresting an Arab demonstrator wearing traditional clothing in Jerusalem in 1948.
Example 5	The picture shows a group of girls, wearing informal clothing, participating in military training on shooting for the Palestinian revolution forces in Beirut, where five of them are carrying rifles and preparing to shoot while the rest are sitting in the background, with a tree behind them.

Table 11: English translations of captions generated by the models under the LLM stacking ensemble configuration.

Averroes at ImageEval 2025 Shared Task: Advancing Arabic Image Captioning with Augmentation and Two-Stage Generation

Mariam Saeed[1,2], Sarah Elshabrawy[3], Abdelrahman Hagrass[1,3],
Mazen Yasser[1,2], Ayman Khalafallah[1,2]

[1]Applied Innovation Center, [2]Alexandria University [3]Georgia Institute of Technology
Applied Innovation Center [m.saeed,a.hagrass,m.yasser,a.khalafallah]@aic.gov.eg
Alexandria University: [es-mariamzaho4,es-mazen2215,ayman.khalafallah]@alexu.edu.eg
Georgia Tech: [selshabrawy3,ahagrass3]@gatech.edu

Abstract

Image captioning aims to generate natural language descriptions of images, combining visual understanding with language generation. This task is particularly challenging in low-resource settings such as Arabic, where annotated data is limited and captions must reflect both cultural and linguistic nuances. In this system paper, we present our approach for the ImageEval 2025 Arabic Image Captioning Shared Task. Our system is based on the Qwen2.5-VL-7B vision-language model, enhanced with quality-aware data augmentation, a two-stage description-to-caption pipeline, and post-processing for improved fluency. In the official evaluation, our approach ranked **first** in the *LLM as a Judge* metric with a score of **33.97**, **second** in *Cosine Similarity* with a score of **58.55**, and **first** in the manual evaluation phase conducted by the organizers.

1 Introduction

Image captioning generates natural language descriptions of images by combining visual understanding with language generation. While vision-language models (VLMs) have achieved strong results in high-resource languages, applying them to Arabic remains challenging due to limited annotated data, complex morphology, and the need for culturally appropriate captions.

The ImageEval 2025 Arabic Image Captioning Shared Task (Bashiti et al., 2025) addressed these challenges by releasing a manually annotated Arabic captioning dataset and a standardized evaluation framework. Systems were evaluated using BLEU (Papineni et al., 2002), Cosine Similarity, and LLM-as-a-Judge scores (Li et al., 2024) during the submission phase, followed by a manual evaluation by the organizers.

We present our system for this task, built on the Qwen2.5-VL-7B model (Team, 2025) with quality-aware data augmentation, a two-stage description-to-caption pipeline, and regex-based post-processing. We also explored lighter models such as BLIP, but Qwen2.5-VL-7B proved superior. Our system ranked **first** in LLM-as-a-Judge (33.97), **second** in Cosine Similarity (58.55), and **first** in manual evaluation, demonstrating the effectiveness of combining large VLMs with targeted augmentation and structured generation for Arabic captioning.

The rest of the paper is organized as follows: Section 3 details our system, Section 4 presents the dataset, metrics, and results, and Section 6 concludes.

2 Related Work

Image captioning aims to produce natural language descriptions of images by combining visual recognition with language generation. Early approaches paired CNN-based encoders with RNN decoders (Vinyals et al., 2015; Karpathy and Fei-Fei, 2017), later enhanced by attention mechanisms (Xu et al., 2015; Anderson et al., 2018) and, more recently, transformer architectures (Cornia et al., 2020).

The field has since shifted toward large vision-language models (VLMs) that integrate powerful image encoders with pretrained language models, enabling stronger cross-modal reasoning. Prominent examples include CLIP (Radford et al., 2021), BLIP (Li et al., 2022), Flamingo (Alayrac et al., 2022), LLaVA (Liu et al., 2023), and Qwen-VL (Bai et al., 2023; Team, 2025), which leverage large-scale multimodal pretraining and instruction tuning to achieve state-of-the-art performance.

Due to their size, adapting VLMs for specific tasks often relies on parameter-efficient fine-tuning methods such as LoRA (Hu et al., 2022), implemented in frameworks like PEFT (Mangrulkar et al., 2022), which significantly reduce computational and memory requirements while preserving model quality.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 432–437
November 8-9, 2025 ©2025 Association for Computational Linguistics

Evaluating image captioning systems has traditionally relied on automatic metrics such as BLEU (Papineni et al., 2002), which measure n-gram overlap, and cosine similarity with TF–IDF (Sparck Jones, 1988; Salton and Buckley, 1988), which captures content similarity beyond surface form. More recently, human-aligned evaluation methods such as LLM-as-a-Judge (Li et al., 2024) have gained attention, assessing captions on semantic accuracy, fluency, and cultural relevance in a manner closer to human judgment.

3 System Overview

In this section, we outline the progression of our experiments during the shared task, starting from initial baselines and gradually introducing more advanced augmentation strategies and post-processing techniques. While we kept the underlying model architecture unchanged, our approach evolved from a single-model setup to a two-model pipeline for improved performance.

3.1 Baseline: Single-Stage Captioning

We began by fine-tuning **Qwen2.5-VL-7B** [1] using LoRA to assess its ability to generate Arabic image captions without any additional enhancements. LoRA allowed us to update only a small subset of parameters while keeping most of the model frozen, reducing computational cost while adapting it to the task dataset. The baseline training prompt was intentionally simple:

> Baseline Prompt
>
> Describe the image in Arabic.

The organizers released scores for both a fully fine-tuned Qwen model and a zero-shot baseline. Our LoRA-based variant yielded different outcomes, which we detail in the results section, and served as the reference point for all subsequent enhancements.

3.2 Smaller Architectures

We wanted to explore the feasibility of using smaller vision-language models for the task, so we experimented with BLIP (Li et al., 2022). We started from a checkpoint already fine-tuned on Flickr8k Arabic captioning dataset[2] and further fine-tuned it on the task dataset. Although BLIP converged quickly, its performance, particularly in capturing fine-grained Arabic details, was noticeably worse than Qwen2.5-VL-7B. Based on these results, we decided to focus on Qwen2.5-VL-7B for the subsequent experiments.

3.3 Data Augmentation Strategies

Given the limited size of the training dataset, we employed two augmentation strategies to improve generalization and assess the performance of different training pipelines.

Aug1: Classical Image Augmentation. The first approach, **Aug1**, applied three random transformations to each image from a predefined set implemented in the `Albumentations` library. The transformations included cropping or padding, horizontal flipping, rotation, small-scale shifting and zooming, motion blur, and Gaussian noise. Captions were kept unchanged, tripling the dataset size and exposing the model to more varied visual patterns while preserving semantic content.

Aug2: Quality-Aware Caption and Image Augmentation. While Aug1 increased visual diversity, it did not introduce textual variation. In **Aug2**, we first augmented captions: for each image, we used **Aya-Vision-8B**[3] to generate three slightly different captions and computed their BLEU score against the original. Captions scoring below 0.75 were discarded to ensure semantic consistency. For each retained augmented caption, one random Aug1 transformation was applied to its image. This process added 814 high-quality samples to the training set. Later, we combined these augmentation strategies with different training pipelines.

3.4 Structured Caption Generation with Descriptions

We hypothesized that guiding the model to first produce a detailed description of the image would lead to more accurate captions. To train such a system, we first created a dataset of image–description–caption triples using **Aya-Vision-8B**. The descriptions were generated with the following prompt:

> Description Data Prompt
>
> Describe this image in detail in Arabic to help in extracting the following caption between <cap> tags.
> <cap>
> caption
> </cap>

[1] https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct

[2] https://huggingface.co/omarsabri8756/blip-Arabic-flickr-8k

[3] https://huggingface.co/CohereLabs/aya-vision-8b

This prompt was designed to produce not only a general description, but also to highlight the key details and important elements that would support accurate caption generation.

We then fine-tuned **Qwen2.5-VL-7B** using a structured output format that explicitly separated the description from the caption:

```
Structured Training Prompt

Describe the image in detail in Arabic.
Then, based on that description, generate
a suitable caption in Arabic (10-30 words).
<description>
Detailed description
</description>
<cap>
Caption
</cap>
```

This structured approach encouraged the model to first organize its observations and reasoning before producing the final caption.

3.5 Two-Model Pipeline

Building on the structured captioning idea, we developed a two-model pipeline. The first model (*Description Model*) generated a detailed description from the image, while the second model (*Caption Model*) used both the image and the description to produce the final caption. Both models were based on Qwen2.5-VL-7B and trained independently.

3.6 Post-Processing and Model Merging

During evaluation, we found that some generated captions contained repetitions or redundant phrases. We applied a regex-based cleaning step to remove such artifacts, improving fluency and readability.

We also observed that two variants of the two-model pipeline excelled in different aspects of captioning, specifically, the pipeline trained with Aug2 and the one without augmentation. To combine their strengths, we performed model merging, a technique that integrates parameters from multiple trained models into a single model, aiming to retain beneficial knowledge from each. We used **MergeKit** (Goddard et al., 2024) with the **TIES** algorithm (Yadav et al., 2023) to merge the models at the parameter level, preserving their complementary capabilities.

3.7 Final System

Our final submission integrated the most effective components from our experiments. It used the **Aug2** quality-aware augmentation to enrich both visual and textual diversity, followed a two-model Qwen2.5-VL-7B pipeline for structured generation (Description → Caption), and applied regex-based cleaning to improve output fluency. The complete workflow is illustrated in Figure 1.

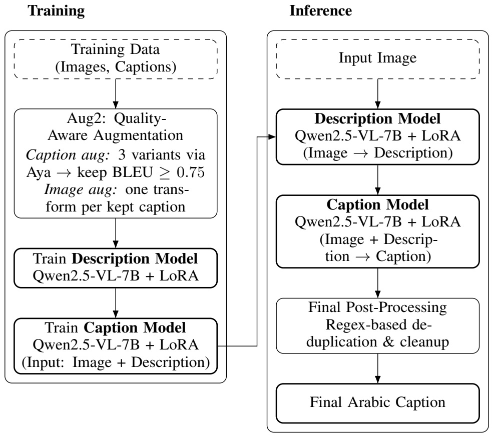


Figure 1: Final system: Aug2 data preparation, two-model Qwen2.5-VL pipeline (Description then Caption), and regex-based cleanup.

4 Experiments

4.1 Dataset and Metrics

We used the training data provided by the organizers of the shared task (Bashiti et al., 2025), a manually captioned dataset of 3,471 images, split into 2,718 for training and 753 for testing. To validate and analyze our approaches during development, we further divided the training set into a 90/10 split, using the smaller portion as a validation set.

Submissions were evaluated using four metrics. During the submission period, **BLEU**, **Cosine Similarity**, and **LLM as a Judge** scores were reported on the public leaderboard. BLEU measured n-gram overlap between the generated caption and the reference, capturing surface-level similarity in wording. Cosine similarity measured the textual closeness between generated captions and reference descriptions after Arabic-specific normalization and TF–IDF vectorization. The LLM-as-a-Judge metric used `gpt-4o` with a fixed seed and zero temperature to score captions on semantic accuracy, relevance, and fluency, with results normalized to a 0–100 scale.

After the submission period, the organizers conducted a **Manual Evaluation** on about 5% of the test set, assessing Cultural Relevance, Conciseness, Completeness, and Accuracy, each on a 1–4 scale.

4.2 Training Setup

All experiments were conducted on a single NVIDIA A100 GPU with 80 GB of memory. The Qwen models were fine-tuned using LoRA with rank 8, targeting all modules. Training was performed with a batch size of 2 and gradient accumulation over 8 steps, giving an effective batch size of 16. For the BLIP model, a batch size of 8 was used. We set the learning rate to 2×10^{-5} with a cosine scheduler and a warmup ratio of 0.1, using `bf16` precision. The input cutoff length was fixed at 2048 tokens. Models were trained for a maximum of 10 epochs, and the checkpoint achieving the lowest loss on our validation set was selected for submission.

4.3 Results and Analysis

Table 1 presents the results of the different variants of our system across BLEU, Cosine Similarity, and LLM-as-a-Judge. The baselines provided by the organizers include a zero-shot Qwen2.5-VL-7B and a fully fine-tuned version. Our LoRA baseline already surpassed both organizer-provided baselines, achieving 22.84 BLEU and 30.19 LLM-as-a-Judge.

Classical image augmentation (**Aug1**) applied to the LoRA baseline slightly reduced LLM-as-a-Judge and Cosine Similarity scores, suggesting that random visual perturbations without textual augmentation do not consistently help. Applying Aug1 to BLIP yielded lower scores overall, confirming that BLIP was less competitive for this task.

Structured output improved semantic evaluation, with the structured-only variant achieving 32.96 LLM-as-a-Judge. Adding quality-aware augmentation (Aug2) increased BLEU to 23.76 but slightly reduced LLM-as-a-Judge, indicating a trade-off between n-gram overlap and semantic quality.

The **two-model pipeline** proved particularly effective, achieving the highest BLEU (24.99) among our systems without augmentation and 33.81 LLM-as-a-Judge when combined with **Aug2**. Merging two-model pipelines trained with and without Aug2 preserved strong BLEU and Cosine scores but slightly lowered LLM-as-a-Judge.

Our final system, two-model pipeline with Aug2 and regex-based output cleaning, achieved the highest LLM-as-a-Judge score (33.97), second place in Cosine Similarity (58.55), and competitive BLEU (24.39), confirming the benefit of structured generation, quality-aware augmentation, and light post-processing.

Model	BLEU	Cosine	LLM-as-a-Judge
Baseline zero-shot (organizers)	9.92	55.77	27.11
Baseline full (organizers)	16.89	58.46	30.82
Baseline LoRA	22.84	56.95	30.19
Baseline + Aug1	22.50	56.33	28.58
BLIP + Aug1	19.95	54.42	19.83
Structured output + Aug2	23.76	57.33	31.71
Structured output	23.31	58.23	32.96
Two-model pipeline	**24.99**	57.72	32.25
Two-model pipeline + Aug2	24.22	**58.58**	33.81
Merging two-model variants	24.55	58.53	33.11
Final system	24.39	58.55	**33.97**

Table 1: Performance of different approaches on the shared task test set.

4.4 Qualitative Analysis

While automatic metrics such as BLEU provide a numerical indication of similarity, qualitative inspection offers a clearer view of how the model succeeds and where it fails. We selected four representative samples from the validation set. The first row of Figure 2 presents two failure cases: in the first, the prediction is entirely off-topic, describing a political exhibition instead of sculptures; in the second, the model confuses a youth music rehearsal with a militant training scene.

The second row shows two successful cases. In the first, the model correctly identifies a military training camp, differing only slightly in phrasing, which still yields a high BLEU score. In the second, the prediction matches the ground truth exactly for a school celebration, resulting in a perfect score. These examples indicate that the model performs reliably on visually distinctive and well-defined scenes but struggles when different events share overlapping contextual cues.

4.5 Official Leaderboard Results

Table 2 shows the official Subtask 2 leaderboard for Cosine Similarity and LLM-as-a-Judge. Our team (**Averroes**) ranked **first** in LLM-as-a-Judge with a score of 33.97 and **second** in Cosine Similarity (58.55).

In the manual evaluation phase (Table 3), our system obtained the highest scores in Cultural Relevance (3.63) and Conciseness (3.43), securing **first place** in this stage of the competition.

5 Limitations

While our approach demonstrates promising results, it is constrained by the limited size of the dataset (2,718 samples), which reduces the model's ability to capture diverse semantic and visual patterns. The augmentation strategies we introduced

Failure 1	
GT (AR)	منحوتات تجسد القرآن والعهد القديم
GT (EN)	Sculptures depicting the Qur'an and the Old Testament
Pred (AR)	صورة للجناح الفلسطيني في معرض دمشق الدولي عام ١٩٧١، برفقة شخصيات من منظمة التحرير الفلسطينية.
Pred (EN)	An image of the Palestinian wing at the Damascus International Fair in 1971, accompanied by figures from the Palestine Liberation Organization. (BLEU = 0.0)

Failure 2	
GT (AR)	صورة يظهر فيها تدريب فريق موسيقي من الشباب الفلسطيني استعداداً للمشاركة في مهرجان الشباب العربي في الجزائر عام ١٩٧٢
GT (EN)	An image showing the training of a Palestinian youth music group in preparation for participating in the Arab Youth Festival in Algeria in 1972
Pred (AR)	صورة لأعضاء من الجبهة الشعبية للتحرير الفلسطيني، داخل أحد مراكز التدريب في بيروت.
Pred (EN)	An image of members of the Popular Front for the Liberation of Palestine inside a training center in Beirut. (BLEU = 0.2094)

Success 1	
GT (AR)	صورة لتدريب جنود جيش التحرير الفلسطيني في أحد معسكرات التدريب
GT (EN)	An image of the training of Palestinian Liberation Army soldiers in one of the training camps
Pred (AR)	صورة لجنود جيش التحرير الفلسطيني في أحد معسكرات التدريب
Pred (EN)	An image of Palestinian Liberation Army soldiers in one of the training camps. (BLEU = 0.7954)

Success 2	
GT (AR)	صورة لحفل مدرسي بمناسبة فلسطينية في إحدى مدارس الفتيات بالكويت.
GT (EN)	An image of a school celebration on a Palestinian occasion in a girls' school in Kuwait.
Pred (AR)	صورة لحفل مدرسي بمناسبة فلسطينية في إحدى مدارس الفتيات بالكويت.
Pred (EN)	An image of a school celebration on a Palestinian occasion in a girls' school in Kuwait. (BLEU = 1.0)

Figure 2: Qualitative examples of Arabic–English captioning. Top row: failure cases with low BLEU scores, where predicted captions diverge from the ground truth. Bottom row: successful cases with high BLEU scores and strong semantic alignment.

Team	Cosine Similarity	LLM-as-a-Judge
VLCAP	**60.01**	33.05
Averroes (ours)	58.55	**33.97**
Phantom Troupe	57.48	31.43
ImpactAi	56.22	26.55
Codezone Research Group	38.30	15.14

Table 2: Official Subtask 2 leaderboard for Cosine Similarity and LLM-as-a-Judge.

Team	Cultural Relevance	Conciseness	Completeness	Accuracy
Averroes (ours)	**3.63**	**3.43**	2.60	2.80
Phantom Troupe	3.40	3.27	2.33	2.40
VLCAP	2.57	3.17	**2.67**	**2.97**
Codezone Research Group	1.10	2.03	1.47	2.03
ImpactAi	3.13	2.73	1.77	1.97

Table 3: Manual evaluation scores on 5% of the test set (1=lowest, 4=highest).

mitigate this limitation to some extent, but cannot fully substitute for a larger, more representative dataset.

Another limitation lies in the reliance on synthetic captions. Although we applied quality control to ensure semantic consistency, automatically generated captions may still introduce noise or overlook subtle aspects of the images.

Finally, our experiments were conducted with a single model size (Qwen2.5-7B). The effect of scaling the model or exploring alternative architectures on caption quality remains an open question for future work.

6 Conclusion

We presented our Qwen2.5-VL-7B–based system for the ImageEval 2025 Arabic Image Captioning Shared Task, integrating quality-aware augmentation, a two-stage description-to-caption pipeline, and regex-based post-processing. The system ranked **first** in *LLM-as-a-Judge*, **second** in *Cosine Similarity*, and **first** in manual evaluation, highlighting the effectiveness of combining large vision-language models with targeted augmentation and structured generation. Future work will explore scaling to larger datasets, multilingual pretraining, and RLHF for improved human alignment.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Ahlam Bashiti, Alaa Aljabari, Hadi Hamoud, Md. Rafiul Biswas, Bilal Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouani. 2025. ImageEval 2025: The First Arabic Image Captioning Shared Task. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10578–10587.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Andrej Karpathy and Li Fei-Fei. 2017. Deep visual-semantic alignments for generating image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):664–676.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *Preprint*, arXiv:2412.05579.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523.

Karen Sparck Jones. 1988. *A statistical interpretation of term specificity and its application in retrieval*, page 132–142. Taylor Graham Publishing, GBR.

Qwen Team. 2025. Qwen2.5-vl.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164.

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2048–2057. JMLR.org.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Preprint*, arXiv:2306.01708.

AZLU at ImagEval Shared Task: Bridging Linguistics and Cultural Gaps in Arabic Image Captioning

Sarah Yassine[*]
Lebanese University / Lebanon
sarah.yassine.2@st.ul.edu.lb

Sara Mahrous
Al Azhar University / Egypt
mahroussara299@gmail.com

Rawan Sous
Birzeit University / Palestine
1200129@student.birzeit.edu

Abstract

Image captioning is the task of automatically generating natural language descriptions for visual content, with applications in search, social media, and beyond. While English captioning has advanced significantly, Arabic captioning remains underdeveloped due to a scarcity of high-quality, culturally relevant datasets. This work, conducted under the ImageEval 2025 Shared Task, addresses this gap by introducing a novel, manually annotated, open-source dataset for Arabic image captioning. Our curated resource consists of 500 unique black-and-white historical photographs documenting pivotal events in modern Palestinian and Lebanese history. The dataset spans from the British Colonial era in Palestine through the events of 1948, and includes documentation of the 1982 Israeli invasion of Beirut. This contribution provides a foundational resource to advance research in Arabic NLP and multimodal systems, offering a vital benchmark for models processing complex historical, cultural, and traumatic imagery.

1 Introduction

Despite significant progress in English image captioning, Arabic captioning remains an understudied challenge due to the language's complex morphology, dialectal diversity, and cultural nuances. These linguistic and contextual gaps hinder the development of robust captioning systems for Arabic content. A key issue identified in recent work on Arabic image captioning is the lack of a well-structured, high-quality dataset in Modern Standard Arabic (MSA) (Mohamed et al., 2023).

To address this gap, we participated in the ImageEval 2025 Shared Task (Bashiti et al., 2025), curating a high-quality dataset of 500 manually annotated images. Our approach enforces strict guidelines: captions are written exclusively in MSA, adhere to length constraints, and prioritize cultural relevance. This ensures consistency and broad usability for both native speakers and learners, while providing a reliable resource for fine-tuning Arabic-capable LLMs.

Crafting high-quality captions in Modern Standard Arabic (MSA) required meticulous precision to accurately describe culturally specific elements (e.g., Palestinian villages, Beirut streets). While regional dialects are prevalent, we employed MSA to ensure broad comprehensibility and establish a formal benchmark. Resources like The Living Arabic Project were leveraged to ensure linguistic accuracy and contextual relevance.

The system achieved strong performance, excelling in automated (41.53 LLM judge score) and human evaluations for conciseness (3.44) and accuracy (3.16). While semantic alignment was lower (59.15 cosine similarity), the methodology proved effective for generating concise, culturally and linguistically accurate Arabic captions.

In summary, our work makes the following contributions:

- A high-quality MSA caption dataset: Carefully annotated with no dialectal influence.
- Culturally relevant descriptions: Each caption reflects the cultural context of the image.
- Linguistically robust system: Our approach ensures factual and grammatical correctness

[*] Corresponding author.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 438–442
November 8-9, 2025 ©2025 Association for Computational Linguistics

in generated captions.

2 Background

2.1 Task Setup

For the ImageEval 2025 Shared Task (Subtask 1: Image Captioning Datathon), our methodology was designed to generate captions that explicitly address the primary evaluation criteria of linguistic quality and cultural relevance.

Input: Our input consisted of a collection of 500 uncaptioned images depicting Palestinian heritage (e.g., traditional villages, daily life) and Beirut during pivotal historical moments (e.g., the Lebanese Civil War, Israeli invasions).

Output: Our generated captions were designed to be culturally grounded in MSA, adhering to the following key constraints. First, a strict word count of between 10 and 50 words. Second, linguistic rigor was maintained by permitting only MSA, excluding regional dialect variants(e.g., اللوح/teaching board/ (al-lawh)), instead of (e.g., السبورة/blackboard/ as-sabbūrah, a regional colloquialism) (Za'ter and Talafha, 2022). Third, we mandated cultural precision. This required the use of specific, contextually appropriate vocabulary, as in the following example: صورة من داخل القدس تظهر المقوش والهندسة الداخلية للمدينة التي تعج بأهلها الذين يرتدون لباسهم التقليدي من عقال وحطة/ (*šūra min dāhil al-quds tazhar al-maqwash wa-l-handasa al-dāhiliyya li-l-madina allati ta'ij bi-ahlihā alladh*^{i}*na yartad*^{u}*na libāsahum al-taqł*^{i}*d*^{i} *min 'iqāl wa-a*a) / (A view from Jerusalem's Old City showing its arched alleys and bustling crowds adorned in traditional keffiyehs and headbands).

2.2 Dataset Details

We manually annotated all 500 images with captions adhering to the above constraints. The dataset is divided into two primary batches of 250 images, each further split into four thematic subsets of 50 images.

2.3 Track Participation

Our participation in Subtask 1 (Image Captioning Datathon) (Bashiti et al., 2025) aimed to demonstrate high-quality caption generation under strict linguistic and cultural constraints. We contributed a rigorously annotated dataset to benchmark MSA compliance and highlight the importance of accurate historical and cultural context over generic descriptions.

3 System Overview

3.1 Design Rationale and Core Principles

Our captioning system addresses critical gaps in prior Arabic image captioning research. Previous work suffered from a scarcity of high-quality public datasets in MSA (Za'ter and Talafha, 2022), a tendency to generate generic descriptions lacking culturally significant details (Emami et al., 2022), and complications from Arabic's dialectal diversity that hinder linguistic consistency (Emami et al., 2022).

To overcome these issues, we implemented a controlled framework. Our primary objective was to create a high-quality, open-source MSA dataset to directly address its scarcity. Consequently, we enforced strict MSA usage, which involved replacing dialectal terms (e.g., سطل, *satl* 'bucket') with their MSA equivalents (دلو, dalw, Bucket).

3.2 Annotation Guidelines

To ensure consistency and quality, all annotators adhered to a strict set of rules.

First, the Language Standard required captions to be written exclusively in MSA, prohibiting dialectal terms to prevent linguistic interference.

Second, the Descriptive Depth guideline mandated comprehensive narratives of 10–50 words, avoiding simple labels (e.g., مسجد, masjid, mosque).

Third, Content Requirements obliged annotators to contextualize scenes by describing precise locations (e.g. المسجد الأقصى في البلدة القديمة بالقدس (*al-Masjid al-Aqsā fī al-balda al-qadīma bi-al-Quds*) 'Al-Aqsa Mosque in the Old City of Jerusalem'), actions, and cultural significance.

Fourth, standardized terminology was achieved by requiring annotators to source all terms from a project glossary validated using The Living Arabic Project dictionary (Living Arabic Project, n.d.).

Finally, Cultural and Historical Accuracy was ensured by verifying culturally significant terms (e.g. يافا (*Yāfā*) 'Jaffa') against historical and multilingual sources, including Wikipedia and digital archives.

3.3 Annotation Challenges and Resolution Strategies

The annotation process encountered several challenges, resolved through structured protocols: First, dialectal interference arises from the team's diverse dialects (e.g., using شروال, shirwāl for trousers). This was mitigated by developing a collaborative glossary to reach consensus on standard MSA terms (e.g. بنطلون, bantalōn, Pants). Second, politically and Culturally Sensitive Terminology required precise language for historical scenes. A mandatory consultation process with historical advisors was instituted to standardize terms (e.g. مجزرة, (majzara), massacre; النضال الفلسطيني (*al-nidāl al-filastīnī*) 'the Palestinian struggle'; الإحتلال البريطاني (*al-ihtilāl al-britānī*) 'the British occupation'). Third, Ambiguity in Transliterated Toponyms was addressed by implementing a verification protocol cross-referencing official maps and historical documents to confirm modern standard Arabic terms (e.g. طبريا (*Tabariyyā*) 'Tiberias').

3.4 Quality Assurance and Validation

A multistage validation process ensured the accuracy of captions. Geographic landmarks were verified against contemporary images from Wikipedia and official records. A linguist also performed random spot checks on finalized captions to verify adherence to all guidelines in Section 3.2.

4 Dataset

4.1 Dataset Overview

This dataset is based on resources from the Shared Task organizers, which we have significantly extended.

First, the organizers provided a core set of 500 uncaptioned images, each with a basic contextual note. The images were organized into ten thematic groups of 50 images.

Second, our contribution was to transform this into a vision and language dataset. We manually authored a relevant and descriptive caption in MSA for eachmage.

Finally, the complete data set contains 500 image caption pairs. Thematic coverage includes: Palestinian resistance (40%), Palestinian cities and villages (30%), events from the Palestinian-Zionist conflict (10%), Beirut during the Lebanese Civil War (10%), and Palestinian daily life and culture (10%).

4.2 Linguistic Analysis of Captions

To characterize the linguistic properties of our manually authored captions, we conducted a quantitative analysis of lexical and syntactic features. This provides a clear profile of the dataset for future users.

Lexical Diversity and Terminology: A frequency analysis of the corpus confirms its thematic focus. The most frequently named entities are location names, led by فلسطين (*Filastīn*) (Palestine, occurring in 32% of captions), القدس (*al-Quds*) (Jerusalem, 28%), and المسجد الأقصى (*al-Masjid al-Aqsā*) (Al-Aqsa Mosque, 15%). As the dataset documents historical events, conceptual nouns such as شهداء (*shuhadā'*) (martyrs) and مجازر (*majāzir*) (massacres) are also highly prevalent, appearing in approximately 35% and 25% of all captions, respectively. The name بيروت (*Bayrūt*) (Beirut) occurs in roughly 10% of the captions, aligning with its defined thematic share.

Syntactic Properties: The captions vary in length from 8 to 50 words, with an average length of 15 words, providing substantive descriptions. A manual analysis of a 100-caption sample revealed that approximately 60% utilize a nominal sentence structure (e.g., الجملة الاسمية (*al-jumla al-ismiyya*), which is typical of descriptive Arabic. Furthermore, given the historical nature of the images, the past tense is the predominant verbal form, used in over 80% of captions that contain a verb.

4.3 Quality Assurance Framework

To ensure the quality and reliability of the captions, we employed a multi-faceted evaluation strategy using automated metrics and human assessment. A detailed analysis of the evaluation results is presented in Section 5 (Results).

4.4 External Tools/Libraries

To ensure factual accuracy, toponym spellings and historical context were verified using Wikipedia and digital archives. This standardized Arabic transliteration against alternative names (e.g., يافا for Jaffa/Yafo), prevents ambiguous geographic references.

5 Results

5.1 Quantitative Results: Ranking and Performance of Official Metrics

In Subtask 1 of ImageEval 2025, performance was assessed through three complementary approaches:

First, Semantic Alignment (Cosine Similarity): Captions were evaluated by computing the average pairwise cosine similarity between TF-IDF vectorized character 3-grams of candidate and reference texts after text normalization. This metric quantifies lexical overlap, accounting for Arabic morphological variation. As shown in Table 1, BZU-AUM led (65.53), while our team (AZLU) scored 59.15.

Second, Automated Quality Assessment (LLM as Judge): A GPT-4o model evaluated captions on a 0–100 scale for semantic accuracy, fluency in MSA, and cultural relevance. Using fixed parameters and a structured prompt ensured reproducibility. Our team (AZLU) led this metric (41.53), reflecting strengths in coherence and relevance, while BZU-AUM scored 32.42.

Third, Manual Evaluation: A 5% stratified sample was evaluated by native Arabic speakers on four qualitative metrics (rated 1–4): Cultural relevance, Concise, Completeness, and Accuracy.

Table 1

Rank (Cosine)	Participants	Cosine Similarity Mean	Rank (LLM)	LLM Judge Score
1	BZU	65.53	2	32.42
2	AZLU	59.15	1	41.53

Table 1: Cosine Similarity and LLM Judge Score results for participating teams.

Table 1 reveals a performance inversion: BZU-AUM led in Cosine Similarity (65.53) but scored lower on the LLM Judge (32.42), while our team (AZLU) led on the LLM Judge (41.53) despite a lower Cosine score (59.15), highlighting a divergence between metric-based and qualitative evaluation.

5.2 Results of the Human Evaluation

The captions were evaluated by a human based on four main criteria: accuracy, completeness, conciseness, and cultural relevance.

Evaluation by native speakers yielded strong scores across key qualitative metrics: Cultural Relevance (3.20), demonstrating effective conveyance of cultural context; Conciseness (3.44), indicating direct and succinct phrasing; Accuracy (3.16), confirming factual alignment with image content. The lower Completeness score (2.88) suggests occasional omissions of finer contextual details.

5.3 Analysis: The Impact of Design Choices

Our captioning system prioritized the exclusive use of Modern Standard Arabic (MSA) to ensure linguistic coherence and prevent dialectal variation. Emphasis was placed on achieving succinctness while preserving cultural and historical accuracy. This methodology generated accurate and concise captions, with strengths in both conciseness and accuracy contributing to strong overall performance.

Participants	Cultural Relevance	Conciseness	Completeness	Accuracy
BZU	3.24	2.76	3.08	2.92
AZLU	3.20	3.44	2.88	3.16

Table 2: Human evaluation results for Subtask 1.

Human evaluation (Table 2) reveals a trade-off: BZU excels in Cultural Relevance and Completeness, while AZLU scores higher in Concise and Accuracy, suggesting a contrast between contextual nuance and precise succinctness.

5.4 Error Analysis: System Mistakes, Confusion Matrices, Error Types

Despite strong overall performance, several areas for improvement were identified:

Linguistic Errors: Occasional use of non-standard MSA terms due to dialectal interference.

Cultural Errors: Due to the absence of location data in some images and limited knowledge of specific locales, unidentified places are designated as 'Palestine'.

Visual Understanding Errors: Challenges in interpreting fine-grained, culturally nuanced scene details.

The evaluation was conducted on the whole dataset after collecting the whole captions in one csv file containing the batch id,image id, and the written caption.

5.5 Distinction between Official vs. Post-Submission Results

Based on the preliminary assessment, our team (AZLU) performed well, particularly in Conciseness and Accuracy, demonstrating the capacity to generate understandable and culturally relevant captions.

6 Limitations

While the Dataset addresses a significant gap in Arabic image captioning resources, it possesses a limitation that presents an opportunity for future work. The dataset's scale, with 500 image-caption pairs, is sufficient for initial benchmarking but remains limited for training large-scale models from scratch without significant data augmentation or transfer learning. A larger-scale dataset would be necessary to achieve state-of-the-art performance and improve model generalization.

7 Conclusion

This paper detailed a contribution to the ImageEval 2025 Shared Task: a manually annotated dataset of Arabic image captions in Modern Standard Arabic (MSA). By enforcing strict linguistic guidelines and prioritizing cultural relevance, we addressed key challenges in Arabic captioning, such as dialectal variation and a lack of public datasets. Our results demonstrated strong performance in conciseness and accuracy, validating the annotation methodology.

While some errors in cultural disambiguation and completeness were observed, this dataset provides a foundational resource. Future enhancements could include expanding the dataset size, integrating multimodal pre-training, and leveraging domain-specific lexicons. This work aims to advance Arabic natural language generation and foster greater inclusion of underrepresented languages in global research.

Acknowledgments

The authors thank the organizers of the ImageEval 2025 shared task, Mustafa Jarrar, Ahlam Bashiti, and Alaa Aljabari, for their work and guidance. We also acknowledge the image annotators.

References

Ahlam Bashiti, Alaa Aljabari, Hadi Hamoud, Md. Rafiul Biswas, Bilal Shalash, Mustafa Jarrar, Fadi Zaraket, George Mikros, Ehsaneddin Asgari, and Wajdi Zaghouani. 2025. ImageEval 2025: The First Arabic Image Captioning Shared Task. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

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*, pages 40–51, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Living Arabic Project. n.d. Living arabic project. https://www.livingarabic.com/.

Abdelrahman Mohamed, Fakhraddin Alwajih, El Moatez Billah Nagoudi, Alcides Inciarte, and Muhammad Abdul-Mageed. 2023. Violet: A vision-language model for Arabic image captioning with gemini decoder. In *Proceedings of ArabicNLP 2023*, pages 1–11, Singapore (Hybrid). Association for Computational Linguistics.

Muhy Eddin Za'ter and Bashar Talafha. 2022. Benchmarking and improving arabic automatic image captioning through the use of multi-task learning paradigm. *Preprint*, arXiv:2202.05474.

Iqra'Eval: A Shared Task on Qur'anic Pronunciation Assessment

Yassine El Kheir
DFKI

Amit Meghanani
University of Sheffield

Hawau Olamide Toyin
MBZUAI

Nada Almarwani
Taibah University

Omnia Ibrahim
Alexandria University

Youssef Elshahawy
HUMAIN

Mostafa Shahin
University of New South Wales

Ahmed Ali
HUMAIN

Abstract

We present the findings of the first shared task on Qur'anic pronunciation assessment, which focuses on addressing the unique challenges of evaluating precise pronunciation of Qur'anic recitation. To fill an existing research gap, the **Iqra'Eval 2025 shared task** introduces the first open benchmark for Mispronunciation Detection and Diagnosis (MDD) in Qur'anic recitation, using Modern Standard Arabic (MSA) reading of Qur'anic texts as its case study. The task provides a comprehensive evaluation framework with increasingly complex subtasks: error localization and detailed error diagnosis. Leveraging the recently developed QuranMB benchmark dataset along with auxiliary training resources, this shared task aims to stimulate research in an area of both linguistic and cultural significance while addressing computational challenges in pronunciation assessment.

1 Introduction

The field of Computer-Aided Pronunciation Training (CAPT) and its core component, Mispronunciation Detection and Diagnosis (MDD), have become indispensable tools for self-directed language learners globally (Neri et al., 2008; Rogerson-Revell, 2021). CAPT systems have two main usages: (*i*) pronunciation assessment, where the system is concerned with the errors in the speech segment; (*ii*) pronunciation teaching, where the system is concerned with correcting and guiding the learner to fix mistakes in their pronunciation (Kheir et al., 2023a). Arabic presents unique challenges for CAPT due to its linguistic complexity and diverse varieties. The Arabic phonological system comprises 34 phonemes, including 28 consonants and 6 vowels with distinct short and long forms, which already surpasses the complexity of many Indo-European languages. A particularly salient challenge is posed by complex phonetic structures not commonly found in other languages, such as uvular and pharyngeal consonants, and the subtle but semantically crucial distinction between emphatic and non-emphatic consonants (e.g., / t/ vs. /T/ or /s/ vs. /S/). A slight mispronunciation, such as a substitution between these pairs, can alter the meaning of a word entirely (Kheir et al., 2023b, 2024; Alrashoudi et al., 2025). These challenges in Arabic are **amplified in the domain of Qur'anic recitation**. The recitation of the Holy Qur'an is governed by a strict set of rules known as Tajweed, which dictates the precise articulation of every phoneme, including specific rules for elongation, nasalization (Idgham, Ikhfaa, Iqlab), and bouncing sounds (Qalqala). These rules introduce a layer of phonetic complexity that is absent in Modern Standard Arabic (MSA) and requires specialized models and datasets that can capture these fine-grained acoustic details (Ahmad et al., 2018; Alagrami and Eljazzar, 2020; Rahman et al., 2021; Alsahafi and Asad, 2024). The IqraEval 2025 challenge is motivated by the Unified Benchmark for Arabic Pronunciation Assessment, with Qur'anic recitation as its case study (El Kheir et al., 2025). Building on this foundation, IqraEval 2025 introduces a standardized benchmark supported by carefully curated datasets to tackle the challenges of Arabic MDD. To fill existing gaps, we present the first open benchmark for mispronunciation detection in MSA, specifically focusing on Qur'anic recitation. Our main contributions are:

- **Task Description:** Quranic Mispronunciation Detection and Diagnosis System.
- **Phoneme Set Description:** Detailed phoneme inventory for MSA-based recitation.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 443–452
November 8-9, 2025 ©2025 Association for Computational Linguistics

- **Dataset Release:** Over 80 hours of training and development speech data.
- **Evaluation Framework:** Clearly defined criteria for benchmarking performance.
- **Leaderboard:** The first public leaderboard for Qur'anic Mispronunciation Detection.

2 Iqra'Eval 2025

2.1 Task Description

The Iqra'Eval 2025 shared task focuses on mispronunciation detection and diagnosis in Qur'anic recitation. Given a speech segment and its corresponding reference transcript, the objective is to automatically identify pronunciation errors and localize their positions. In this first iteration of the shared task, the task is framed as a phoneme recognition problem, where the systems are expected to accurately predict the pronounced phonemes in a given MSA-style read Qur'anic Arabic speech recording.

2.2 Dataset and Evaluation

2.2.1 Training Dataset

CV-Ar Dataset This dataset incorporates an 82.37 hours subset of the Common Voice Dataset (Ardila et al., 2019) version 12.0 specifically for MSA Arabic speech recognition. The data set consists of read speech samples collected from a diverse pool of speakers with a well-balanced gender distribution. Fully vowelized versions of the transcriptions developed by El Kheir et al. were used in this shard task. Additionally, the corpus has been augmented with samples drawn from Qur'anic recitations (Alrashoudi et al., 2025).

TTS Augmentation Dataset Introduced in El Kheir et al. to address the scarcity of mispronounced annotated speech data, based on the generative approach techniques demonstrated in Korzekwa et al. 2022. The authors used seven in-house single-speaker TTS systems (5 male and 2 female voices) trained on fully vowelized transcriptions to generate 26 hours of error-free speech (canonical pronunciations) and 26 hours of speech with systematically introduced mispronunciations. The mispronunciation patterns were created by systematically modifying the input of canonical transcripts based on a predefined confusion-pairs matrix derived from phoneme similarity data extracted from Kheir et al. 2022.

2.2.2 Testing Dataset

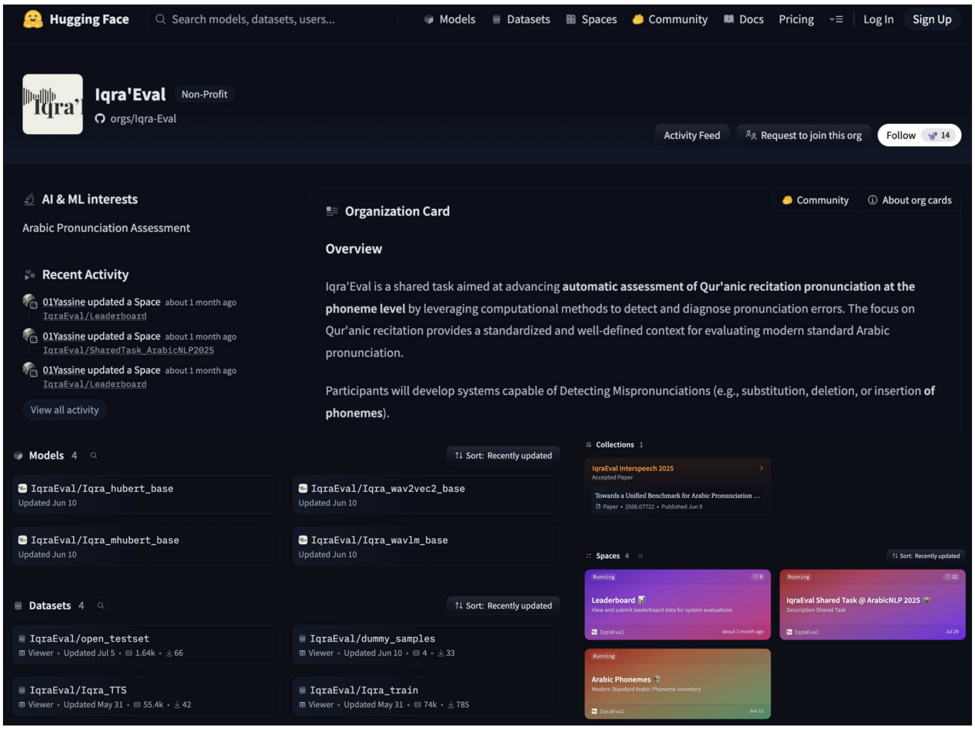
QuranMB The test set introduced by El Kheir et al. consists of 98 verses from the Qur'an, recited by 18 native Arabic speakers (14 females, 4 males), resulting in approximately 2.2 hours of recorded speech. The speakers were instructed to read the text while deliberately producing specified pronunciation errors, which were systematically selected to emulate the most prevalent mispronunciations reported in the literature on common errors in Qur'anic recitation. A custom recording tool was developed to highlight modified text and display additional instructions specifying the type of error to ensure consistency in error production (Alrashoudi et al., 2025). The test set was further annotated by 3 Arabic linguistic annotators.

2.3 Evaluation

We utilize **the specialized phoneme set for Qur'anic Arabic** developed in El Kheir et al., 2025, which builds on the phonetizer introduced by Halabi and Wald, 2016. This set of phonemes includes 62 unique phonemes that account for all MSA sounds, including gemination (the doubling of consonant sounds). The phonemizer has been optimized for phonetic coverage in speech synthesis, employing a greedy algorithm to minimize corpus size while maintaining comprehensive phonetic and prosodic coverage.

Evaluation Metrics Our evaluation protocol adopts the hierarchical structure established in prior mispronunciation detection research (Li et al., 2016; Leung et al., 2019; Kheir et al., 2023a). This framework jointly considers (*i*) the annotated verbatim sequence, (*ii*) the canonical text-dependent reference sequence, and (*iii*) the model prediction. Based on the alignment of these three sources, predictions are categorized into four primary classes:

- **True Accept (TA):** correctly accepted phones that are both annotated and predicted as correct pronunciations.
- **True Reject (TR):** correctly rejected phones that are both annotated and predicted as mispronunciations. These are further exploited to distinguish between *Correct Diagnosis (CD)* and *Error Diagnosis (ED)* depending



Iqra'Eval Non-Profit

orgs/Iqra-Eval

AI & ML interests

Arabic Pronunciation Assessment

Recent Activity

01Yassine updated a Space about 1 month ago
IqraEval/Leaderboard

01Yassine updated a Space about 1 month ago
IqraEval/SharedTask_ArabicNLP2025

01Yassine updated a Space about 1 month ago
IqraEval/Leaderboard

Organization Card

Overview

Iqra'Eval is a shared task aimed at advancing **automatic assessment of Qur'anic recitation pronunciation at the phoneme level** by leveraging computational methods to detect and diagnose pronunciation errors. The focus on Qur'anic recitation provides a standardized and well-defined context for evaluating modern standard Arabic pronunciation.

Participants will develop systems capable of Detecting Mispronunciations (e.g., substitution, deletion, or insertion **of phonemes**).

Models 4

IqraEval/Iqra_hubert_base
Updated Jun 10

IqraEval/Iqra_wav2vec2_base
Updated Jun 10

IqraEval/Iqra_mhubert_base
Updated Jun 10

IqraEval/Iqra_wavlm_base
Updated Jun 10

Datasets 4

IqraEval/open_testset
Viewer • Updated Jul 5 • 1.64k • 66

IqraEval/dummy_samples
Viewer • Updated Jun 10 • 4 • 33

IqraEval/Iqra_TTS
Viewer • Updated May 31 • 55.4k • 42

IqraEval/Iqra_train
Viewer • Updated May 31 • 74k • 785

Collections 1

IqraEval Interspeech 2025
Accepted Paper

Towards a Unified Benchmark for Arabic Pronunciation …
Paper • 2506.07722 • Published Jun 9

Spaces 4

Leaderboard 📊
View and submit leaderboard data for system evaluations
IqraEval about 1 month ago ♡ 9

IqraEval Shared Task @ ArabicNLP 2025
Description Shared Task
IqraEval Jul 29 ♡ 12

Arabic Phonemes 📚
Modern Standard Arabic Phoneme Inventory
IqraEval Jun 11

Figure 1: Iqra'Eval Shared Task main page.

on whether the predicted phone matches the canonical pronunciation.

- **False Reject (FR):** phones that are actually correct but are incorrectly predicted as mispronunciations.
- **False Accept (FA):** phones that are actually mispronounced but misclassified as correct.

From these four categories, we derive the following error rates.

$$FRR = \frac{FR}{TA + FR} \tag{1}$$

$$FAR = \frac{FA}{FA + TR} \tag{2}$$

$$DER = \frac{ED}{CD + ED} \tag{3}$$

In addition to error rates, we adopt standard diagnostic metrics to evaluate system performance. Precision and Recall are defined as:

$$\text{Precision} = \frac{TR}{TR + FR} \tag{4}$$

$$\text{Recall} = \frac{TR}{TR + FA} = 1 - FAR \tag{5}$$

Finally, the overall performance is summarized using the F1-score, i.e., the harmonic mean of Precision and Recall:

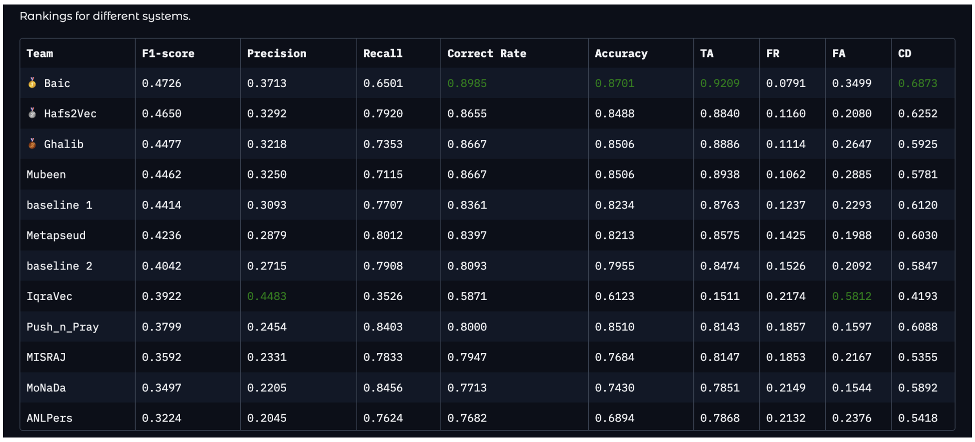
$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

3 Shared Task Teams

Submission Rules All resources for IqraEval are consolidated on the dedicated Hugging Face organization page[1] (see Fig. 1). This page serves as the central hub for datasets, baseline models, reference resources, and evaluation tools. Its main components are summarized as follows:

- **Baseline Models.** Four pretrained SSL models are released for participants: `Iqra_hubert_base`, `Iqra_wav2vec2_base`, `Iqra_mhubert_base`, and `Iqra_wavlm_base`. These provide standardized starting points and ensure comparability across submissions.
- **Datasets.** The page hosts multiple datasets covering training, evaluation, and auxiliary resources:

[1] https://huggingface.co/IqraEval



Rankings for different systems.

Team	F1-score	Precision	Recall	Correct Rate	Accuracy	TA	FR	FA	CD
🥇 Baic	0.4726	0.3713	0.6501	0.8985	0.8701	0.9209	0.0791	0.3499	0.6873
🥈 Hafs2Vec	0.4650	0.3292	0.7920	0.8655	0.8488	0.8840	0.1160	0.2080	0.6252
🥉 Ghalib	0.4477	0.3218	0.7353	0.8667	0.8506	0.8886	0.1114	0.2647	0.5925
Mubeen	0.4462	0.3250	0.7115	0.8667	0.8506	0.8938	0.1062	0.2885	0.5781
baseline 1	0.4414	0.3093	0.7707	0.8361	0.8234	0.8763	0.1237	0.2293	0.6120
Metapseud	0.4236	0.2879	0.8012	0.8397	0.8213	0.8575	0.1425	0.1988	0.6030
baseline 2	0.4042	0.2715	0.7908	0.8093	0.7955	0.8474	0.1526	0.2092	0.5847
IqraVec	0.3922	0.4483	0.3526	0.5871	0.6123	0.1511	0.2174	0.5812	0.4193
Push_n_Pray	0.3799	0.2454	0.8403	0.8000	0.8510	0.8143	0.1857	0.1597	0.6088
MISRAJ	0.3592	0.2331	0.7833	0.7947	0.7684	0.8147	0.1853	0.2167	0.5355
MoNaDa	0.3497	0.2205	0.8456	0.7713	0.7430	0.7851	0.2149	0.1544	0.5892
ANLPers	0.3224	0.2045	0.7624	0.7682	0.6894	0.7868	0.2132	0.2376	0.5418

Figure 2: Iqra'Eval Shared Task Leaderboard.

  - `IqraEval/Iqra_train`: training corpus for system development.
  - `IqraEval/open_testset`: public evaluation split for leaderboard submissions.
  - `IqraEval/Iqra_TTS`: synthetic speech dataset for data augmentation and robustness testing.
  - `IqraEval/dummy_samples`: lightweight set for debugging and format verification.

- **Arabic Phonemes.** A dedicated Space provides an interactive inventory of MSA phonemes, including examples of canonical pronunciations, which supports error diagnosis.

- **Papers.** A collection highlights accepted publications, including the IqraEval Interspeech 2025 paper (El Kheir et al., 2025), which formally describes the benchmark.

- **Leaderboard**[2]**.** An interactive Hugging Face Space is maintained to visualize and compare system outputs. Submitted predictions are automatically evaluated and the leaderboard is updated with human-in-the-loop.

- **Code Samples and Evaluation Scripts.** The organization provides baseline code, sample commands, and the official implementation of evaluation metrics to standardize experimental pipelines and ensure reproducibility.

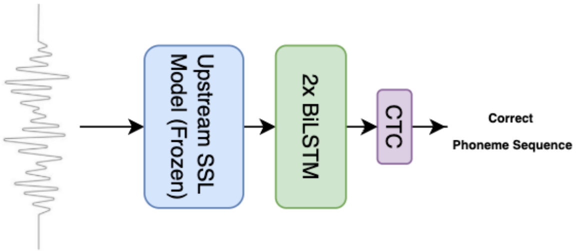


Figure 3: Mispronunciation Detection Modeling Pipeline

- **Submission Workflow.** Participants submit their system outputs in the prescribed CSV format by email following the format provided by the organizers. All valid runs are evaluated automatically, and the results are published on the leaderboard.

Participating Teams A total of 29 teams registered for the shared task. Out of these, 11 teams actively participated in the testing phase and had their systems ranked on the official leaderboard. Among them, 6 teams submitted a system description paper, and 5 of these were accepted for publication in the proceedings of the Iqra'Eval shared task. The participation spanned multiple regions across the globe, including teams from the Middle East, North Africa, Sub-Saharan Africa, South Asia, Europe, North America and Oceania, reflecting the international interest and diversity of the research community engaged in this task.

Baselines We establish two baselines for the Iqra'Eval shared task, as shown in our benchmark (El Kheir et al., 2025), both of which leverage

[2] https://huggingface.co/spaces/IqraEval/Leaderboard

Team	Affiliation	Paper Published
ANLPers	Prince Sultan University, Saudi Arabia	✓
BAIC	Applied Innovation Center , Egypt	✓
Greentech	Greentech Apps Foundation, United Kingdom/Bangladesh	
Hafs2Vec	The University of New South Wales, Australia	✓
IqraVec	Imperial College London, United Kingdom	
Metapseud	Independent, Sudan	✓
Misraj Tech	Misraj Technology, Saudi Arabia	
MONADA	- , Tunisia	✓
Mubeen	- , -	
Ghalib	- , -	
Push_n_Pray	Euromed University of Fes, Morocco	

Table 1: List of teams that participated in Iqra'Eval Shared Task.

SSL speech models combined with temporal modeling, as illustrated in Figure 3. Following the SUPERB setup (wen Yang et al., 2021), the SSL encoder parameters are frozen and the layer-wise representations are aggregated through a weighted sum across transformer layers. The resulting features are passed to a model head consisting of a 2-layer, 1024-unit Bi-LSTM trained with Connectionist Temporal Classification (CTC) (Graves et al., 2006) loss on phoneme sequences. During inference, phoneme sequences are obtained using greedy CTC decoding.

- **Baseline 1 (mHuBERT)**: This system employs the multilingual HuBERT (mHuBERT) (Zanon Boito et al., 2024), pretrained on 90,430 hours of speech covering 147 languages. It represents a strong multilingual SSL model suitable for cross-lingual phoneme recognition.

- **Baseline 2 (WavLM).**: This system is based on WavLM (Chen et al., 2022), a 94M-parameter model pretrained on English speech. It provides a monolingual reference point against the multilingual variant.

Our baselines allow us to contrast the effectiveness of multilingual versus monolingual SSL representations for MDD. Below, we provide a brief description for each team system.

3.1 ANLPers:

ANLPers'System (Qandos et al., 2025) is based on `Whisper-large-v3` (Radford et al., 2023), the largest Whisper model with 1.55B parameters and multilingual capabilities. Audio input is resampled to 16 kHz as required by Whisper. The tokenizer is extended with 68 phoneme tokens from (Halabi and Wald, 2016), and the embedding layer is resized accordingly.

The dataset is preprocessed to retain only audio and phoneme attributes. Audio features are extracted using the Whisper feature extractor, and phoneme sequences are encoded as labels. Training is performed using the Hugging Face `transformers` library with a batch size of 4, gradient accumulation of 4 steps, a learning rate of 1×10^{-5}, and 2 epochs.

3.2 BAIC:

BAIC'System (Mattar et al., 2025) is based on Wav2Vec2-BERT (Chung et al., 2021). It employs task-adaptive continued pretraining on large Arabic speech datasets, using phoneme-level labels automatically generated via the Iqra'Eval phonetizer, followed by fine-tuning on the official training data augmented with synthetic Quran recitations created using XTTS-v2. This strategy allows the model to internalize fine-grained phonetic distinctions relevant to mispronunciation detection.

3.3 Hafs2Vec:

This system (Ibrahim, 2025) was trained on two datasets: EveryAyah/QUL, consisting of 94 hours of Quran recitations from 28 professional reciters (filtered to verses under 10 seconds, 54k clips), and the IqraEval training set (79 hours, 74k clips). Phoneme labels for the reciters were generated using a custom Quranic phonemizer that outputs context- and Tajweed-aware phoneme sequences aligned with the IqraEval phoneme set. The model is based on `facebook/wav2vec2-xls-r-1b` and fine-tuned for 15 epochs with an effective batch size of 352, a learning rate of 3×10^{-5}, AdamW optimization, and CTC loss over the phoneme vocabulary, trained on the UNSW Katana HPC with mixed precision.

3.4 Metapseud:

The submission (Mansour, 2025) applies domain adaptation with multi-stage fine-tuning for phoneme-level Qur'anic mispronunciation detection using Wav2Vec2.0. In the first stage, the pretrained `wav2vec2-large-xlsr-53-arabic` model is fine-tuned on a large Qur'anic phoneme-annotated dataset (245k recitations), producing a general-purpose phoneme recognizer. In the second stage, the model is further fine-tuned on the official IqraEval training set (79h) to specialize in Qur'anic phoneme structures. Decoding is performed with CTC and beam search, which improves performance on the IqraEval open test set.

3.5 MONADA:

Team MONADA (DAOUD and MESSAOUD, 2025) designed a lightweight system to balance performance with memory efficiency by placing a shallow transformer on top of a pretrained Wav2Vec2.0 feature extractor. Raw audio is processed with the S3PRL Wav2Vec2.0 Base featurizer, producing 768-dimensional frame-level representations, which are then projected into a smaller hidden dimension and fed into a 3-layer transformer encoder with 4 attention heads per layer and a feed-forward size of 1024. The model is trained using CTC loss. Training is conducted for 15 epochs with Adam optimizer (learning rate 3×10^{-4}, cosine annealing scheduler, minimum learning rate 1.5×10^{-5}), dropout of 0.15, and gradient clipping. The best model is selected based on correct rate performance on the development set.

3.6 Mubeen:

The system is based on fine-tuning a Whisper-medium model using the IqraEval training and TTS augmentation data. Only the decoder layers were trained, while the encoder was frozen due to limited hardware and time, using a learning rate of 1×10^{-5} for 2 epochs. An additional fine-tuning pass applied a conservative SpecAugment (Park et al., 2019) strategy, and the resulting models were combined by weight averaging. Inference employed three model configurations with a pairwise WER voting strategy.

3.7 Usubmitted Papers

Out of the 11 participating teams in the IqraEval 2025 Shared Task, 5 teams submitted their test set results to the leaderboard but did not provide any system description or accompanying paper. While their performance contributed to the overall competition rankings, the lack of documentation prevents a detailed analysis of their approaches, training strategies, or architectural choices.

4 Shared Task Results

The overall results for the shared task are in Table 2. Team BAIC (Mattar et al., 2025) presented the best approach, with the best score in 5 of 9 metrics reported. Their model is followed closely by Hafs2Vec (Ibrahim, 2025) and Ghalib. The top 2 approaches included additional training data with BAIC using synthetic data and Hafs2Vec using Quranic recitation from human speakers. The Quranic recitation supplementation data (94 hours) might have affected the model's performance since the training set (79 hours) for Iqra'Eval 2025 is read in MSA style.

5 Lessons from the First Quran Pronunciation Challenge

The submissions revealed three main sources of innovation: model design, data strategies, and training/inference practices. These reflect the community's attempt to balance performance, computational cost, and linguistic specificity.

5.1 Model Innovations

Most teams built on large pretrained encoders such as Whisper, Wav2Vec2, XLS-R, or Wav2Vec2-BERT, demonstrating the effectiveness of transfer learning for Qur'anic mispronunciation detection. Some groups explored lightweight designs, for example MONADA and ShallowTransformer, which placed shallow transformer layers on top of frozen representations to reduce computational cost. Other innovations included extending model vocabularies, such as ANLPers, which

Team	F1-score↑	Precision↑	Recall↑	Correct Rate↑	Accuracy↑	TA	FR	FA	CD
Baic	**0.4726**	0.3713	0.6501	**0.8985**	**0.8701**	**0.9209**	0.0791	0.3499	**0.6873**
Hafs2Vec	0.4650	0.3292	**0.7920**	0.8655	0.8488	0.8840	0.1160	0.2080	0.6252
Ghalib	0.4477	0.3218	0.7353	0.8667	0.8506	0.8886	0.1114	0.2647	0.5925
Mubeen	0.4462	0.3250	0.7115	0.8667	0.8506	0.8938	0.1062	0.2885	0.5781
baseline 1	0.4414	0.3093	0.7707	0.8361	0.8234	0.8763	0.1237	0.2293	0.6120
Metapseud	0.4236	0.2879	0.8012	0.8397	0.8213	0.8575	0.1425	0.1988	0.6030
baseline 2	0.4042	0.2715	0.7908	0.8093	0.7955	0.8474	0.1526	0.2092	0.5847
IqraVec	0.3922	**0.4483**	0.3526	0.5871	0.6123	0.1511	0.2174	**0.5812**	0.4193
Push_n_Pray	0.3799	0.2454	0.8403	0.8000	0.8510	0.8143	0.1857	0.1597	0.6088
MISRAJ	0.3592	0.2331	0.7833	0.7947	0.7684	0.8147	0.1853	0.2167	0.5355
MoNaDa	0.3497	0.2205	0.8456	0.7713	0.7430	0.7851	0.2149	0.1544	0.5892
ANLPers	0.3224	0.2045	0.7624	0.7682	0.6894	0.7868	0.2132	0.2376	0.5418
GreenTech	0.1997	0.1128	**0.8682**	0.5033	0.4585	0.5093	**0.4907**	0.1318	0.4719

Table 2: Evaluation results of different submissions across multiple metrics. Best scores per column are highlighted in bold. Dashed line separates the top 3 submissions.

augmented Whisper's tokenizer with 68 Quran-specific phoneme tokens and resized embeddings accordingly.

5.2 Data Innovations

Several submissions showed that carefully designed resources were central to performance. Hafs2Vec introduced a Tajweed-aware phonemizer to capture recitation rules such as Idgham and Ikhfaa, ensuring that phoneme sequences reflected Qur'anic articulation. BAIC and Mubeen demonstrated the value of TTS-based augmentation using XTTS-v2 to generate synthetic recitations. Hafs2Vec also mixed data from EveryAyah/QUL with the official IqraEval training set to expand speaker and style diversity. In addition, BAIC applied large-scale automatic phoneme labeling with the IqraEval phonetizer to enable task-adaptive continued pretraining on Arabic speech corpora.

5.3 Training and Inference Innovations

Beyond data and model design, training practices had a notable impact. Metapseud applied multi-stage fine-tuning, first adapting to a large Qur'anic phoneme corpus and then specializing on IqraEval. Mubeen selectively fine-tuned only the Whisper decoder layers due to hardware constraints, showing a practical path for parameter-efficient adaptation. Other strategies included the use of SpecAugment and conservative regularization, weight averaging, WER-based voting, and beam search decoding. BAIC highlighted the benefits of task-adaptive pretraining, further reinforcing the importance of domain-specific adaptation.

The summary of innovation by each team is described in the Table 3.

5.4 Emerging patterns

A number of common themes emerged across systems. All teams relied on pretrained SSL encoders, either Whisper or Wav2Vec2 variants, underlining their versatility as general-purpose feature extractors. Quran-specific resources, such as Tajweed-aware phonemizers and synthetic recitations, consistently boosted accuracy and provided linguistic grounding. Training strategies such as multi-stage fine-tuning, selective adaptation, and ensembles yielded measurable gains even without major architectural changes. Finally, the leaderboard revealed different trade-offs in recall versus precision, with some systems favoring high recall for error detection and others prioritizing precision for stricter evaluation.

5.5 Iqra'Eval 2025 Limitations

Limited Linguistic Scope The generalizability of our findings is constrained by the limited linguistic scope of the test data. Although the written Quranic text is standardized, modern spoken Arabic exhibits significant dialectal variation. However, for this shared task, test data was collected exclusively from speakers of the Saudi Arabic dialect. This might limit the generalizability of the results, as the model may fail to capture the rich diversity of the Arabic spoken language.

Team/System	Model Innovation	Data Innovation	Training/Inference Innovation
ANLPers	Whisper-large-v3 with extended phoneme tokenizer	–	HF fine-tuning with resized embeddings
BAIC	Wav2Vec2-BERT backbone	TTS augmentation (XTTS-v2); automatic phoneme labeling	Task-adaptive pretraining; fine-tuning on augmented data
Hafs2Vec	Wav2Vec2-XLS-R-1B	Custom Tajweed-aware phonemizer; mixing EveryAyah/QUL + IqraEval	Large-batch CTC training with AdamW; mixed precision
Metapseud	Wav2Vec2.0 (xlsr-53-arabic)	Large Qur'anic phoneme corpus (245k)	Multi-stage fine-tuning; CTC with beam search
MONADA	Lightweight shallow transformer on Wav2Vec2 features	–	Efficient training with cosine annealing, dropout, clipping
Mubeen	Whisper-medium with frozen encoder; decoder-only training	TTS augmentation	SpecAugment; weight averaging; WER-based voting

Table 3: Summary of innovations from teams that participated in the first Iqra'Eval Shared Task, grouped by model, data, and training methods.

Targets-Specific Common Errors During data collection, speakers were instructed to produce specific pronunciation mistakes deliberately. While this covers some common mistakes, it may not accurately represent the subtle, context-dependent errors that occur in natural recitation, which could limit the model's ability to detect errors in real-world scenarios.

Children IqraEval To the best of our knowledge, there are no publicly available corpora dedicated to children's Qur'an pronunciation learning and recitation assessment. This lack of resources highlights a significant research gap, mainly due to the difficulties of collecting and annotating children's recitation data.

6 Future Work:

We propose the following three directions for future research on our challenge, informed by insights from Iqra'25:

6.1 Task Modelling

Looking ahead, three areas appear particularly promising: (*i*) the balance between precision and recall remains an open challenge: systems must avoid over-flagging errors while still catching subtle mispronunciations; (*ii*) resource creation is essential, especially for rare phonemes and Tajweed-specific contexts where current datasets are imbalanced; (*iii*) efficient adaptation methods such as parameter-efficient fine-tuning, streaming-friendly architectures, or lightweight ensembles could make these models more practical for deployment in real learning settings.

6.2 Data Collection

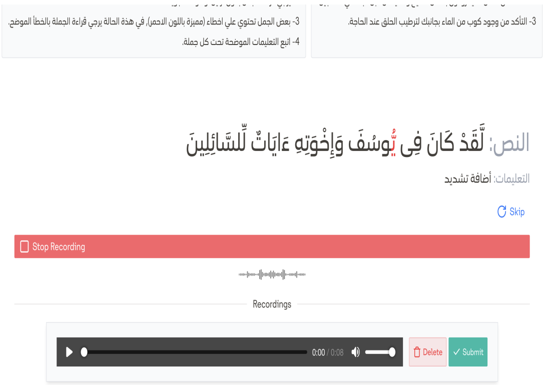
We need more effort to collect and incorporate data from a wide range of Arabic dialects, including but not limited to Egyptian, Levantine, and North African. To capture real-world errors, the next step is to collect recitation audio from a large, diverse group of non-professional reciters, which will then be annotated to identify spontaneous mispronunciations. Further more, we need to develop and release a dedicated corpus for children's Qur'an pronunciation learning and recitation assessment, addressing the current absence of such resources.

6.3 Crowdsourcing platform

We addressed the lack of available data by developing a custom crowd-sourcing platform [3]. This web application allows users to register and provide basic demographic information, including their spoken language, gender, and age. For each

[3] https://quran-data-collection.sanad.ink

sentence, a specific instruction guides the user to introduce a targeted mispronunciation as shown in Figure 4. In cases where a sentence is particularly challenging, no mispronunciation instruction is given, and the user simply reads the sentence as it is. Finally, after a user submits their recordings, we collect the audio data along with their demographic metadata. This information is then prepared for release as a dataset. The collected data will be shared on the Hugging Face platform as part of a shared task, making it accessible to the wider research community. We invite researchers to use our platform to participate and build diveristy corpora for the next Iqra' challenge.



3- التأكد من وجود كوب من الماء بجانبك لترطيب الحلق عند الحاجة.

3- بعض الجمل تحتوي علي اخطاء (مميزة باللون الاحمر), في هذة الحالة يرجي قراءة الجملة بالخطأ الموضح.

4- اتبع التعليمات الموضحة تحت كل جملة.

النص: لَّقَدْ كَانَ فِى يُّوسُفَ وَإِخْوَتِهِ ءَايَاتٌ لِّلسَّائِلِينَ

التعليمات: أضافة تشديد

Figure 4: Data collection screenshot

Acknowledgments

We would like to thank all the annotators who spent hours and hours reviewing our data, and we are grateful to the volunteers who have made lots of recordings across multiple dialects. We thank the Saudi Data & AI Authority (SDAIA) for hosting the Winter School[4] where this work started.

References

Fadzil Ahmad, Saiful Zaimy Yahya, Zuraidi Saad, and Abdul Rahim Ahmad. 2018. Tajweed classification using artificial neural network. In *2018 International Conference on Smart Communications and Networking (SmartNets)*, pages 1–4. IEEE.

Ali M Alagrami and Maged M Eljazzar. 2020. Smartajweed automatic recognition of arabic quranic recitation rules. *arXiv preprint arXiv:2101.04200*.

Norah Alrashoudi, Hend Al-Khalifa, and Yousef Alotaibi. 2025. Improving mispronunciation detection and diagnosis for non-native learners of the arabic language. *Discover Computing*, 28(1):1.

Yousef S Alsahafi and Muhammad Asad. 2024. Empirical study on mispronunciation detection for tajweed rules during quran recitation. In *2024 6th International Conference on Computing and Informatics (ICCI)*, pages 39–45. IEEE.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.

Mohamed Nadhir DAOUD and Mohamed Anouar BEN MESSAOUD. 2025. Phoneme-level mispronunciation detection in quranic recitation using shallowtransformer. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, Salima Mdhaffar, Thomas Hain, Yasser Hifny, Mostafa Shahin, and Ahmed Ali. 2025. Towards a Unified Benchmark for Arabic Pronunciation Assessment: Qur'anic Recitation as Case Study. In *Interspeech 2025*, pages 2410–2414.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, ICML '06, page 369–376, NY, USA. Association for Computing Machinery.

Nawar Halabi and Mike Wald. 2016. Phonetic inventory for an arabic speech corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 734–738.

[4] https://www.aiwinterschool.com/

Ahmed Ibrahim. 2025. Hafs2vec: A system for the iqraeval arabic and qur'anic phoneme-level pronunciation assessment shared task. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Yassine El Kheir, Ahmed Ali, and Shammur Absar Chowdhury. 2023a. Automatic pronunciation assessment–a review. *arXiv preprint arXiv:2310.13974*.

Yassine El Kheir, Shammur Absar Chowdhury, Ahmed Ali, Hamdy Mubarak, and Shazia Afzal. 2022. Speechblender: Speech augmentation framework for mispronunciation data generation. *arXiv preprint arXiv:2211.00923*.

Yassine El Kheir, Fouad Khnaisser, Shammur Absar Chowdhury, Hamdy Mubarak, Shazia Afzal, and Ahmed Ali. 2023b. Qvoice: Arabic speech pronunciation learning application. *arXiv preprint arXiv:2305.07445*.

Yassine El Kheir, Hamdy Mubarak, Ahmed Ali, and Shammur Absar Chowdhury. 2024. Beyond orthography: Automatic recovery of short vowels and dialectal sounds in arabic. *arXiv preprint arXiv:2408.02430*.

Daniel Korzekwa, Jaime Lorenzo-Trueba, Thomas Drugman, and Bozena Kostek. 2022. Computer-assisted pronunciation training—speech synthesis is almost all you need. *Speech Communication*, 142:22–33.

Wai-Kim Leung, Xunying Liu, and Helen Meng. 2019. Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8132–8136. IEEE.

Kun Li, Xiaojun Qian, and Helen Meng. 2016. Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):193–207.

Ayman Mansour. 2025. Metapseud at iqra'eval: Domain adaptation with multi-stage fine-tuning for phoneme-level qur'anic mispronunciation detection. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Bassam Mattar, Mohamed Fayed, and Ayman Khalafallah. 2025. Aras2p: Arabic speech-to-phonemes system. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Ambra Neri, Ornella Mich, Matteo Gerosa, and Diego Giuliani. 2008. The effectiveness of computer assisted pronunciation training for foreign language learning by children. *Computer Assisted Language Learning*, 21(5):393–408.

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. *arXiv preprint arXiv:1904.08779*.

Nour Qandos, Serry Sibaee, Samar Ahmad, OMER NACAR, Adel Ammar, Wadii Boulila, and Yasser Alhabashi. 2025. Anplers at iqraeval shared task: Adapting whisper-large-v3 as speech-to-phoneme for qur'anic recitation mispronunciation detection. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Munirah Ab Rahman, Izatul Anis Azwa Kassim, Tasiransurini Ab Rahman, and Siti Zarina Mohd Muji. 2021. Development of automated tajweed checking system for children in learning quran. *Evolution in Electrical and Electronic Engineering*, 2(1):165–176.

Pamela M Rogerson-Revell. 2021. Computer-assisted pronunciation training (capt): Current issues and future directions. *Relc Journal*, 52(1):189–205.

Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, and Kushal Lakhotia et al. 2021. Superb: Speech processing universal performance benchmark. In *Interspeech 2021*, pages 1194–1198.

Marcely Zanon Boito, Vivek Iyer, Nikolaos Lagos, Laurent Besacier, and Ioan Calapodescu. 2024. mhubert-147: A compact multilingual hubert model. In *Interspeech 2024*, pages 3939–3943.

Hafs2Vec: A System for the Iqra’Eval Arabic and Qur’anic Phoneme-level Pronunciation Assessment

Ahmed Ibrahim
University of New South Wales
ahmed.ibrahim8165@gmail.com

Abstract

This paper details our submission–Hafs2Vec– to the Iqra’Eval 2025 shared task on Arabic mispronunciation detection. Our system is built upon a wav2vec2-xls-r-1b model, enhanced by two key contributions: a strategic data mixing approach and a custom Qur’anic phonemizer. We augment the official Iqra’Eval training data with 94 hours of professional Qur’anic recitations, creating a balanced dataset that combines learner speech with high-quality acoustic references. To accurately label the reciter data, we developed a custom, Tajweed-aware phonemizer that captures the specific articulation rules of Qur’anic recitation. On the QuranMB test set, our system achieved an F1-score of 46.50% and a high recall of 79.20%.

1 Introduction

The Iqra’Eval 2025 shared task (Kheir et al., 2025) provides a crucial benchmark for advancing Computer-Aided Pronunciation Training in the nuanced domain of Modern Standard Arabic (MSA) and Qur’anic recitation. A primary challenge in developing effective mispronunciation detection systems is the inherent variability in speech data. Learner datasets often contain valuable error patterns but may lack acoustic consistency, while professional recordings offer pristine quality but no examples of common mistakes. Bridging this gap is essential for building models that are both robust and accurate.

To address this challenge, our work introduces two primary contributions. First, we employ a data mixing strategy that combines the 79-hour Iqra’Eval training set with 94 hours of professional Qur’anic recitations. This approach is designed to improve the model’s generalisation by exposing it to a wider range of acoustic conditions, speaking styles, and phonetic details, balancing error diversity with acoustic quality. Second, to enable this strategy, we developed a custom Qur’anic phonemizer (Ibrahim, 2025). This tool generates precise phonetic transcriptions for the professional reciter data by incorporating complex Qur’anic articulation rules governed by Tajweed rules and special symbols within the Qur’anic Uthmani script, which are not accounted for in standard MSA phonemizers, such as Halabi and Wald, 2016.

By integrating these components into a fine-tuned self-supervised learning model framework, our system achieves strong performance. This paper details our methodology, from data preparation and phoneme label generation to model training and evaluation, and provides an in-depth analysis of the system’s performance and error patterns.

2 Methodology

2.1 Data Configuration

We train on a mixture of professional and normal recitations to balance acoustic quality with speaker and error diversity. 28 professional reciters from EveryAyah (Anonymous, 2010) and Qur’anic Universal Library (Tarteel, 2025) are used, filtered to verses of length $\leq$ 10s, yielding $\sim$94 h ($\sim$54k utt.).

The Iqra’Eval training set contributes $\sim$79 h ($\sim$74k utt.) of CommonVoice (Ardila et al., 2020) Arabic speech augmented with Qur’anic recitations.

2.2 Phoneme Label Generation

For the Iqra’Eval data, phoneme labels were provided by the organisers. For the professional reciters set, we generated phoneme labels automatically using a custom Qur’anic phonemizer. This tool takes a verse reference as input and outputs a context-aware, Tajweed-aware phoneme sequence. It expands beyond Modern Standard Arabic phonetics by incorporating Tajweed articulation rules, including Idgham, Iqlab, Ikhfaa, and Qalqala. The phoneme inventory used was matched to the of-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 453–456
November 8-9, 2025 ©2025 Association for Computational Linguistics

Dataset	Utterances	Hours	PER (%)	Sub. (%)	Del. (%)	Ins. (%)
Iqra'Eval Dev (All)	2588	3.4	7.69	4.29	1.62	1.78
Iqra'Eval Dev (Qur'an)	615	–	3.88	1.97	0.84	1.07
Iqra'Eval Dev (MSA)	1973	–	9.50	5.39	1.99	2.11
Professional Reciters	1443	2.6	1.55	0.92	0.37	0.26

Table 1: Phoneme Error Rate (PER) and error type breakdown on development sets.

System	TAR↑	FRR↓	FAR↓	CD↑	Recall↑	Precision↑	F1↑
Organisers' baseline	86.21	13.79	24.44	66.78	75.56	17.67	28.64
Leaderboard Winner (Baic)	**92.09**	**7.91**	34.99	**68.73**	65.01	**37.13**	**47.26**
Our system (Hafs2Vec)	88.40	11.60	**20.80**	62.52	**79.20**	32.92	46.50

Table 2: Mispronunciation detection comparison on the QuranMB test set. TAR: True Acceptance Rate, FRR: False Rejection Rate, FAR: False Acceptance Rate, CD: Correct Diagnosis. Values are percentages. ↓ lower is better, ↑ higher is better.

ficial Iqra'Eval phoneme set — mostly through direct mapping from the phonemizer output, alongside some pre-training and post-training rules.

2.3 Training Configuration

We employed an end-to-end system based on the multilingual facebook/wav2vec2-xls-r-1b (Babu et al., 2021), fine-tuned for 15 epochs with an effective batch size of 352 (22 training batch size × 4 gradient accumulation steps × 4 GPUs). Optimization was performed using AdamW with a learning rate of 3e-5 and a warm-up ratio of 0.1. Experiments were conducted on the University of New South Wales Katana high-performance computing cluster with mixed precision. For inference, greedy decoding was used.

3 Results and Analysis

3.1 Development Set Performance

We evaluated the systems on the Iqra'Eval development set, further categorised into Qur'an and MSA only versions, and the professional reciters development set, consisting of 3 unique reciters and unseen training verses. Table 1 summarises development sets, their PER values and error breakdowns.

On the Iqra'Eval development set, the system achieved a PER of 7.69%, with substitutions (4.29%) as the dominant error type, followed by insertions (1.78%) and deletions (1.62%). Performance is notably better on Qur'anic speech (3.88% PER) than on MSA speech (9.50% PER). This is likely due to the significant variation in style between the CommonVoice MSA data and augmented Qur'anic data. That being said, the MSA subset has approximately 3 times the utterances of Qur'anic subset, so its PER is statistically more stable.

The professional reciters development set shows the lowest PER at 1.55%, reflecting the clarity, consistent articulation, and style match to the training data.

3.2 Test Set Performance

Table 2 compares test set performance of our system with 1. the organisers' baseline system (Kheir et al., 2025) using mHuBERT trained on the CMV-Ar data and 2. the leaderboard-winning system.

Our model achieves a high recall (79.20%) and a low false acceptance rate (20.80%), with a competitive F1-score (46.50%), indicating strong coverage of actual mispronunciations while avoiding many incorrect error detections. The winner leads in F1-score (47.26%) and precision (37.13%), reflecting a more conservative error detection strategy that trades some recall for higher precision.

These results highlight a key trade-off: our system favours high recall and balanced acceptance/rejection behaviour, making it suitable for learner-feedback scenarios where missing genuine errors is more costly than flagging occasional false positives. In contrast, the winning system's higher precision may be advantageous in applications prioritising concise, accurate feedback over exhaustive detection.

3.3 Impact of Data Augmentation

Combining the Iqra'Eval data with professional reciter data introduces greater voice diversity, variation in recitation speed (slow to fast), and exposure to a broader range of acoustic conditions, allowing

Category	Errors	Phonemes	Category PER (%)	Overall PER Contribution (%)
Consonant phonemes	2156	44540	4.84	2.57
Vowel phonemes	4044	36491	11.08	4.82
Shaddah phonemes	238	2818	8.45	0.28
Total	**6438**	**83849**	**–**	**7.69**

Table 3: Error breakdown by phoneme category on the Iqra'Eval development set.

the model to generalise to different recitation styles. Quranic recitation follows distinct modes—such as *mujawwad*, *murattal*, and *hadr*—each with its own tempo and melodic features. Unlike general MDD, we argue that for Quranic MDD it is essential that models are robust to these stylistic differences and that PER is evaluated on multiple test sets representing different styles. A significant decrease in one domain's PER could be more valuable to the overall system at the expense of a slight increase in a different domain's PER, which motivates our use of multiple development sets.

Furthermore, the system demonstrated strong ability to differentiate between the two data domains, successfully avoiding false detection of Tajweed phonemes in the test set, where the Qur'an was recited without Tajweed. Specifically, across the 1,643-utterance test set, only 7 Ikhfaa phonemes and 14 Qalqala phonemes were present, with zero false insertions for all other Tajweed rules. This suggests that the model effectively learned to condition its predictions on the presence or absence of Tajweed articulation, rather than overfitting to Tajweed-rich training data.

Further experimentation could explore the optimal mixing strategy between the two domains, for example by adjusting the ratio of learner to reciter data, applying curriculum learning, or selective sampling.

3.4 Model Selection Findings

Internal experimentation showed that the wav2vec2-xls-r-1b model outperformed the 300M-parameter variant by ∼1.3% absolute PER as well as other models of similar size from the HuBERT (Hsu et al., 2021) and WavLM (Chen et al., 2022) families. This outcome is consistent with the expectation that larger parameter counts enable better representation of complex acoustic patterns, such as those found in Arabic and Qur'anic phonemes. It is reasonable to expect that the 2B-parameter xls-r model could yield further improvements; however, the primary limitations are the increased computational requirements for training and the slower inference speed, alongside risks of overfitting to the training data.

3.5 Categorical Error Analysis

Table 3 presents a breakdown of phoneme errors by broad phoneme categories of consonants, vowels and shaddah phonemes on the development set, allowing for a more robust evaluation of the system (Loweimi et al., 2023). In addition to the total error counts (*Errors*) and the total number of reference phonemes in each category (*Phonemes*), the table reports the *Category PER*, computed for that category alone, and the *Overall PER Contribution*, which reflects the contribution of that category to the overall PER of the dataset.

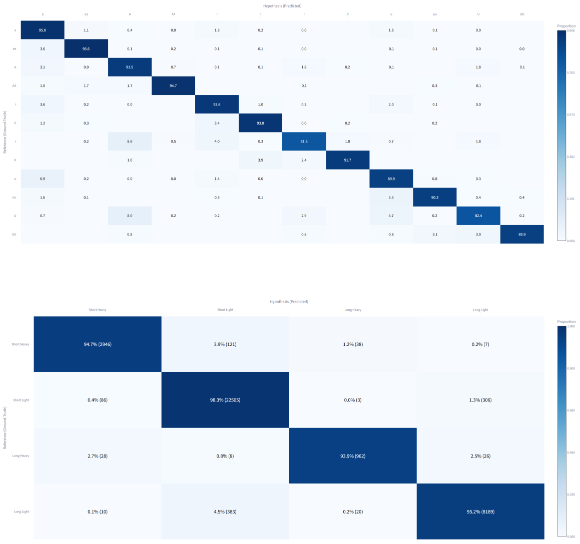


Figure 1: Confusion matrices for vowel phonemes on the development set. (Top) All 12 vowel labels. (Bottom) Vowels grouped by phonetic category: short light = {a u i}, short heavy = {A U I}, long light = {aa uu ii}, long heavy = {AA UU II}.

Vowel phonemes dominate errors, with a PER contribution of 4.82 out of the overall 7.69 (62.8% of all misrecognitions), and exhibiting the highest category PER at 11.08. As seen in the confusion matrices in Figure 1, many of the 12 vowel labels represent acoustically similar sounds (long/short and light/heavy variants), which likely increases confusion, even for human listeners. A possible mitigation would be to reduce the vowel inventory to 6 or 8 broader categories, merging acoustically similar variants while preserving distinctions essential for Arabic speech.

Shaddah (gemination) phonemes, although responsible for only 0.28 PER, have a relatively high category PER (8.45) given their scarcity in the training data: all ten least frequent phonemes in the training data are shaddah forms, each with fewer than 600 occurrences, and the rarest ("EE" and "HH") occur 90 and 94 times respectively. Addressing this severe imbalance may require targeted techniques such as oversampling utterances containing rare phonemes, enforcing balanced phoneme distributions in training batches, or adjusting the loss function to penalise errors on under-represented classes more heavily.

Consonant phonemes are more numerous overall but have a lower category PER (4.84), contributing 33.5% of total errors. These results highlight vowels as the dominant source of phoneme errors, followed by consonants, while rare shaddah phonemes remain a disproportionate challenge given their scarcity in the training data.

4 Conclusion

Our contribution to the Iqra'Eval 2025 shared task demonstrates the effectiveness of a mixed-data training strategy for Arabic mispronunciation detection. By combining learner data and professional, Tajweed-rich recitations with a custom Qur'anic phonemizer, our wav2vec2-xls-r-1b based system is able to generalise across different styles. Our system achieved an F1-score of 46.50%, notable for its high recall (79.20%) on the test set, demonstrating a strong ability to identify genuine errors while minimising incorrect flags.

Our categorical error analysis revealed that acoustically similar vowel phonemes are the primary source of errors ($\sim$63% of all misrecognitions), suggesting that future work focusing on targeted data augmentation or refined vowel inventories could yield significant improvements in the robustness of Arabic pronunciation assessment systems. Furthermore, systematic exploration of optimal data mixing techniques and curriculum learning strategies could further enhance model performance.

5 Acknowledgements

We thank the Iqra'Eval shared task organisers for providing the datasets, evaluation platform, and baseline systems that made this work possible. This research includes computations using the computational cluster Katana supported by Research Technology Services at UNSW Sydney (DOI: 10.26190/669X-A286).

References

Anonymous. 2010. Everyayah dataset. https://everyayah.com/. Online.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. *Preprint*, arXiv:1912.06670.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. XLS-R: self-supervised cross-lingual speech representation learning at scale. *CoRR*, abs/2111.09296.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Nawar Halabi and Mike Wald. 2016. Phonetic inventory for an Arabic speech corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 734–738, Portorož, Slovenia. European Language Resources Association (ELRA).

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *Preprint*, arXiv:2106.07447.

Ahmed Ibrahim. 2025. Quranic phonemizer. https://github.com/Hetchy/Quranic-Phonemizer.

Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Olamide Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, Salima Mdhaffar, Thomas Hain, Yasser Hifny, Mostafa Shahin, and Ahmed Ali. 2025. Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study. *Preprint*, arXiv:2506.07722.

Erfan Loweimi, Andrea Carmantini, Peter Bell, Steve Renals, and Zoran Cvetkovic. 2023. Phonetic error analysis beyond phone error rate. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:3346–3361.

Tarteel. 2025. Quranic universal library (qul) — recitations and segments data. https://qul.tarteel.ai/resources/recitation.

Phoneme-level mispronunciation detection in Quranic recitation using ShallowTransformer

Mohamed Nadhir DAOUD, Mohamed Anouar BEN MESSAOUD
Laboratoire Signal, Images et Technologies de l'Information
Université de Tunis El Manar, Tunis, Tunisia
mohamednadhir@gmail.com, anouar.benmessaoud@yahoo.fr

Abstract

Preserving the integrity of Qur'anic recitation requires accurate pronunciation, as even subtle mispronunciations can alter meaning. Automatic assessment of Qur'anic recitation at the phoneme level is therefore a critical and challenging task. We present ShallowTransformer, a lightweight and computationally efficient transformer model leveraging Wav2vec2.0 features and trained with CTC loss for phoneme-level mispronunciation detection. Evaluated on the Iqra'Eval benchmark (QuranMB.v2), our model outperforms published BiLSTM baselines on QuranMB.v1 while achieving competitive performance relative to the official Iqra'Eval challenge baselines, which are not yet fully documented. Such improvements are particularly important in assisted Qur'an learning, as accurate phonetic feedback supports correct recitation and preserves textual integrity. These results highlight the effectiveness of transformer architectures in capturing subtle pronunciation errors while remaining deployable for practical applications.

1 Introduction

Mispronunciation detection and diagnosis (MDD) systems play a key role in computer-assisted pronunciation training (CAPT), helping language learners identify and correct pronunciation errors without human instructors (Neri et al., 2008). The detection component aims to detect pronunciation anomalies, whereas the diagnosis component aims to assign a specific class to each anomaly.

Most of the foundational research and development of MDD systems has been conducted in the context of English. For example, datasets such as L2-Arctic which includes non-native English speech annotated at phoneme level, for substitution, insertion, and deletion errors, have been extensively used to train and benchmark detection algorithms (Jiang et al., 2021).

In contrast, progress in mispronunciation detection for low-resource languages such as Arabic has been slow. The Arabic phonological system contains 28 consonants and 6 vowels (short and long), where complex phonetic structures (for example, uvular and pharyngeal sounds) (Alotaibi and Muhammad, 2010), present unique problems that do not commonly arise in more standardized and highly resourced languages. Moreover, subtle phonetic contrasts, such as between emphatic and non-emphatic consonants, can be difficult to perceive (Alrashoudi et al., 2025).

Furthermore, the diversity of Arabic dialects introduces substantial variability in pronunciation and vocabulary, while code-switching further complicates speech modeling efforts (Besdouri et al., 2024). These factors, along with the absence of short-vowel diacritics in most written text, create unique challenges for both learners and automated pronunciation assessment systems.

Previous research on Arabic mispronunciation detection has relied on either simplistic datasets such as isolated letters (Ziafat et al., 2021) or words (Aly et al., 2021), or on privately collected corpora that are not publicly accessible (Nazir et al., 2019)(Algabri et al., 2022). This reliance on private and limited datasets has prevented the establishment of standardized benchmarks and hindered objective comparison between different methodologies. (El Kheir et al., 2025) recently released an open phoneme annotated Arabic dataset, designed to provide a unified benchmark for Arabic pronunciation assessment. Building on the release of this benchmark dataset, we are positioned to rigorously evaluate advanced mispronunciation detection methods.

We present an end-to-end Arabic MDD model that leverages self-supervised speech representations. Our approach uses a pretrained wav2vec 2.0 encoder (Baevski et al., 2020) to extract robust acoustic features from raw audio, followed

Proceedings of The Third Arabic Natural Language Processing Conference, pages 457–463
November 8-9, 2025 ©2025 Association for Computational Linguistics

by a shallow Transformer network (Vaswani et al., 2017) trained with a Connectionist Temporal Classification (CTC) loss (Graves et al., 2006) to predict phoneme sequences. This combination enables the system to learn fine-grained phonetic distinctions while avoiding the need for explicit phonetic alignments.

Our contributions are:

- **Model:** A phoneme-level Arabic MDD system combining wav2vec 2.0 acoustic representations with a lightweight Transformer encoder trained via CTC.

- **Dataset:** An evaluation of the proposed approach on the QuranMB.v1 dataset (Kheir et al., 2025).

- **Analysis:** A performance comparison against baseline approaches, including an error-type breakdown to assess diagnostic capabilities for different phonetic categories.

2 Related Works

Earlier mispronunciation detection (MDD) methods primarily used the Goodness of Pronunciation (GOP) metric (Witt and Young, 2000), an objective measure of pronunciation quality based on likelihood scores. GOP computes the likelihood of acoustic segments corresponding to each phoneme using a set of Hidden Markov Models (HMMs).

(Harrison et al., 2009) used a GMM-HHM acoustic model to extract phone level representations. Phonological rules are modeled with finite state transducer to create an extended recognition network (ERN). This approach requires modeling correct pronunciation but also common mispronunciations.

(Li et al., 2016) overcame the need to design mispronunciation rules in ERN, by using a deep neural network that predict L2-speaker pronunciation from acoustic features and canonical phonemes, allowing for simultaneous detection and diagnosis of pronunciation anomalies.

CTC-CNN-RNN was introduced in (Leung et al., 2019) to leverage the ability of convolutional neural networks (CNN) to extract features, recurrent neural networks (RNN) to model sequences and CTC-loss to avoid explicit alignment between input frames and target phoneme sequence.

(Wu et al., 2021) used an encoder-decoder type transformer to predict phones from MFCC features and conduct experiments on the CU-CHLOE corpus (Meng et al., 2007).

The success of large language models in natural language processing, not only showed the power of scaling transformer models, but revealed also the importance of self-supervised learning (SSL) as a pre-training technique. This is no different for speech tasks, where it has been proven that transformer models can learn self-supervised speech representations (SSSR) (Mohamed et al., 2022).

Foundation models such as Wav2Vec 2.0 (Baevski et al., 2020) and HuBERT (Hsu et al., 2021) became widely used for SSSR extraction.

(Peng et al., 2021) finetuned Wav2Vec 2.0 on the TIMIT dataset (Garofolo et al., 1993) to then test it on L2-Arctic. While (Wu et al., 2021) used Wav2Vec 2.0 as a backbone to extract SSSR and use it as input to an MLP prediction layer.

MDD for arabic was also influenced by the same trends, wher for example (Algabri et al., 2022) used CNN-RNN-CTC technique on Arabic-CAPT, a private dataset that contains phoneme transcription of Arabic words. Also (Alrashoudi et al., 2025) finetuned Wav2Vec 2.0 and HuBERT on the L2-KSU data set. While (Kheir et al., 2025) uses frozen SSL models as backbones for SSSR extraction, to train a BiLSTM based model.

Our work builds on these trends by employing a Wav2Vec 2.0 encoder for feature extraction and a shallow transformer for phoneme prediction, enabling accurate detection and diagnosis of mispronunciations in Arabic speech while balancing performance with memory efficiency.

3 Methodology

We propose a shallow transformer-based approach for Arabic phoneme sequence recognition. Our architecture leverages pre-trained wav2vec2 features and a lightweight transformer encoder. We opt for a shallow transformer to balance accuracy with computational efficiency. This approach makes the model easily deployable on resource-constrained environments, such as mobile applications or embedded systems used by learners. The model is trained end-to-end using CTC loss for automatic alignment.

3.1 Datasets

3.1.1 Training and dev sets

The CMV-Ar data corpus, detailed in (Kheir et al., 2025), is derived from the Common Voice Dataset

(Ardila et al., 2019) and enhanced with Quranic recitation samples. It includes a training set of 71,391 utterances (approximately 79 hours of speech) and a development set of 2,588 utterances (3.33 hours of raw audio). Each audio file in the corpus is accompanied by its corresponding spoken phoneme sequence.

3.1.2 Test set

(Kheir et al., 2025) utilized the QuranMB.v1 test set, which contains 2.2 hours of Qur'anic recitation from 18 native Arabic speakers, the majority of whom are female. A more recent release, QuranMB.v2, was made publicly available through the Iqra'Eval challenge (El Kheir et al., 2025). This updated version includes 98 utterances from the same 18 speakers, totaling approximately 2 hours of audio, although the exact differences between the two versions remain unclear. The corresponding labels for QuranMB.v2 are not yet available; however, performance metrics can be obtained by submitting predicted phoneme sequences to an online API.

All sets labels are based on the phoneme dictionary provided by (Halabi and Wald, 2016).

3.2 Audio feature extraction with Wav2Vec 2.0

3.2.1 Wav2Vec 2.0

Proposed by (Baevski et al., 2020), Wav2Vec 2.0 is a self-supervised framework for learning speech representations. It learns contextualized audio features from raw waveforms. The model consists of a convolutional feature encoder, which transforms audio signals into latent representations, and a Transformer-based context network, which captures long-range temporal dependencies.

During pretraining, Wav2Vec 2.0 uses a contrastive loss to predict masked latent representations from their surrounding context. This enables the model to learn rich, domain-agnostic acoustic representations without requiring transcriptions. It has been proven that these representations can be fine-tuned for various downstream tasks or used directly as high-quality feature vectors, thereby reducing the need for large datasets.

3.2.2 Featurizer

The authors of (Kheir et al., 2025) provided several pretrained models as baselines, that can be loaded using the S3PRL toolkit (Yang et al., 2024). We used the pretrained `iqra_wav2vec2_base`[1] checkpoint to load the upstream feature extractor , which returns features extracted by 13 layers of the pretrained Wav2Vec 2.0. The default S3PRL featurizer computes a weighted sum of these 13 representations for each frame.

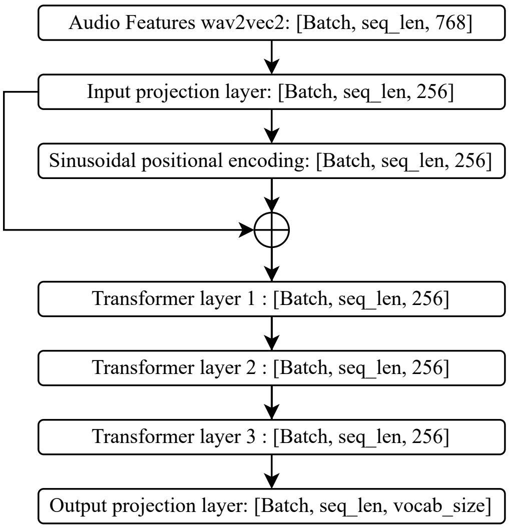

Audio Features wav2vec2: [Batch, seq_len, 768]
Input projection layer: [Batch, seq_len, 256]
Sinusoidal positional encoding: [Batch, seq_len, 256]
Transformer layer 1 : [Batch, seq_len, 256]
Transformer layer 2 : [Batch, seq_len, 256]
Transformer layer 3 : [Batch, seq_len, 256]
Output projection layer: [Batch, seq_len, vocab_size]

Figure 1: Architecture of the Shallow Transformer.

3.3 CTC loss

Introduced by (Graves et al., 2006), CTC loss allows for aligning speech utterances with associated shorter phoneme sequences without requiring explicit alignments. Instead of forcing a one-to-one correspondence between input frames and output labels, CTC loss allows for repetitions and blank symbols in the predicted sequence. This enables the model to handle variations in speaking speed and pronunciation, as well as silence between phonemes. The loss function sums the probabilities of all valid alignment paths that correspond to the true phoneme sequence, effectively allowing the model to learn the most probable sequence without needing pre-segmented data.

3.4 Detailed architecture of ShallowTransformer

ShallowTransformer (ST), depicted in Figure 1, incorporates three stacked transformer layers. To optimize training performance and memory consumption, we downsample the audio features from 768 to 256. We augmented the input data with sinusoidal positional encoding. Subsequently, an output linear layer transforms the 256 transformer encodings to match the vocabulary size. The model's out-

[1] https://huggingface.co/IqraEval/Iqra_wav2vec2_base

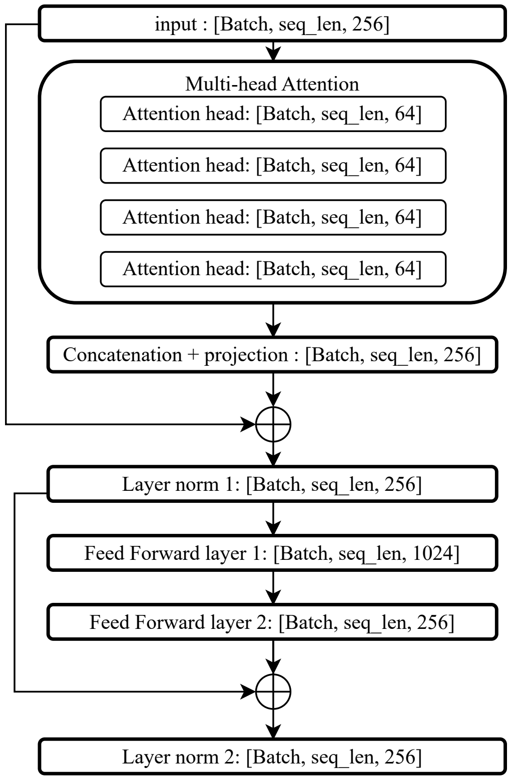

input : [Batch, seq_len, 256]
Multi-head Attention
Attention head: [Batch, seq_len, 64]
Attention head: [Batch, seq_len, 64]
Attention head: [Batch, seq_len, 64]
Attention head: [Batch, seq_len, 64]
Concatenation + projection : [Batch, seq_len, 256]
Layer norm 1: [Batch, seq_len, 256]
Feed Forward layer 1: [Batch, seq_len, 1024]
Feed Forward layer 2: [Batch, seq_len, 256]
Layer norm 2: [Batch, seq_len, 256]

Figure 2: Architechture of transformer layers.

put comprises logits with dimensions [Batch size, sequence length, vocabulary size]. These logits are utilized by the CTC loss for loss computation and by a decoding algorithm to produce the predicted sequence. In order to process speech features in batches, all samples are padded to the length of the sample with maximum length.

As depicted in Figure 2 each transformer layer has 4 attention heads, followed by a shared layer normalization layer and a feed forward network, that projects the 256-dim features to 1024 (4 x 256) and back again to 256.

3.5 Tokenization

Phoneme-level tokenization was performed using the provided phoneme vocabulary. A blank token was added to the vocabulary at index 0, which is necessary for CTC loss. Each phoneme's token corresponds to its index.

3.6 CTC decoding and post-processing

The model outputs are processed using argmax at each time step to obtain the most likely token sequence. Before applying CTC decoding rules, predictions are truncated to actual sequence lengths to ignore padding tokens. The CTC alignment is then collapsed by applying two standard rules:

Model	Recall	Precision	F1-score
BiLSTM (Wav2vec2)	76.72	15.71	26.08
BiLSTM (WavLM)	75.35	15.80	26.12
BiLSTM (HuBERT)	74.75	15.67	25.91
BiLSTM (mHuBERT)	75.56	17.67	28.64
ST (Wav2vec2)	**84.56**	**22.05**	**34.94**

Table 1: Recall, precision, and F1-score of the proposed model compared to published baselines on the QuranMB.v1 test set

1. merging consecutive identical non-blank tokens into single occurrences.
2. removing all blank tokens.

This greedy approach provides efficient decoding, making it suitable for real-time phoneme recognition applications.

3.7 Metrics

The used metrics follow the established MDD convention defined in (Qian et al., 2010). This approach classifies predictions into four groups: True Accept (TA), False Accept (FA), True Reject (TR) and False Reject (FR). Precision, recall and F1-score are then computed following:

$$\text{Recall} = \frac{\text{TR}}{\text{TR} + \text{FA}} \tag{1}$$

$$\text{Precision} = \frac{\text{TR}}{\text{TR} + \text{FR}} \tag{2}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

3.8 Training configuration

The Adam optimizer was employed with a learning rate of 3e-4. A Cosine annealing scheduler was used, setting the minimum learning rate at 1.5e-5. Regularization included Dropout at 0.15 and gradient clipping with a maximum norm of 1.0. Training was conducted for 15 epochs, including 3 warmup epochs, with a batch size of 64 samples.

4 Experimental Results and Comparative Analysis

Table 1 reports the performance of our Shallow Transformer (ST) model on the QuranMB.v1 benchmark in comparison with previously published baselines from (Kheir et al., 2025). Across all three evaluation metrics—recall, precision, and

Model	Recall	Precision	F1-score
baseline 1 (IqraEval)	77.07	**30.93**	**44.14**
baseline 2 (IqraEval)	79.08	27.15	40.42
ST (Wav2vec2)	**84.56**	22.05	34.94

Table 2: Recall, precision, and F1-score of the proposed model compared to the official Iqra'Eval shared task baselines on the QuranMB.v2 test set.

F1-score—our model outperforms the BiLSTM-based baselines using different SSL feature extractors. The largest improvement is observed in F1-score, where our model achieves 34.94% compared to the best baseline score of 28.64%. Although these results indicate a substantial performance gain, it should be noted that QuranMB.v1 and QuranMB.v2 are not identical. While they are similar in duration and number of speakers, the exact differences are not documented. As such, direct numerical comparison should be interpreted with caution.

Table 2 presents our results on the QuranMB.v2 dataset alongside the baselines provided by the organizers of the Iqra'Eval shared task. These baselines serve as strong reference points for this test set, although they have not yet been officially published or fully documented.

Our model achieves the highest recall (84.56%) among all compared systems, but lower precision and F1 score than both baselines. This suggests that while our model is highly sensitive in detecting relevant phoneme events, further optimization is needed to improve precision and overall balance between recall and precision. Nonetheless, the results confirm the competitiveness of our approach under the same evaluation protocol.

5 Conclusion

We presented ShallowTransformer, a lightweight model for automatic phoneme level assessment of Qur'anic recitation pronunciation, leveraging self-attention on SSL-based acoustic features and trained with CTC loss. The model was designed to balance accuracy with computational efficiency, making it suitable for practical deployment. Our results show substantial improvements over published BiLSTM baselines: 10% higher recall, 25% higher precision, and over 22% higher F1-score, while remaining competitive with the official Iqra'Eval challenge baselines.

Although precision remains lower than recall, indicating a higher rate of false rejects, ShallowTransformer demonstrates strong capability in capturing pronunciation patterns. Future work will focus on improving precision through refined decoding, richer data augmentation, and exploring more advanced model architectures.

Acknowledgments

We would like to express our sincere gratitude to the organizers of the Iqra'Eval shared task — Yassine El Kheir (DFKI - Technical University of Berlin), Amit Meghanani (University of Sheffield), Mostafa Shahin (University of New South Wales), Omnia Ibrahim (Alexandria University), Hawau Olamide Toyin (MBZUAI), Nada Al-Marwani (Taibah University), Youssef Elshahawy (HUMAIN), and Ahmed Ali (HUMAIN) — for their invaluable efforts in designing and coordinating this initiative aimed at advancing the automatic assessment of Qur'anic recitation pronunciation.

References

Mohammed Algabri, Hassan Mathkour, Mansour Alsulaiman, and Mohamed A Bencherif. 2022. Mispronunciation detection and diagnosis with articulatory-level feedback generation for non-native arabic speech. *Mathematics*, 10(15):2727.

Yousef Ajami Alotaibi and Ghulam Muhammad. 2010. Study on pharyngeal and uvular consonants in foreign accented arabic for asr. *Computer Speech & Language*, 24(2):219–231.

Norah Alrashoudi, Hend Al-Khalifa, and Yousef Alotaibi. 2025. Improving mispronunciation detection and diagnosis for non-native learners of the arabic language. *Discover Computing*, 28(1):1.

Salah A Aly, Abdelrahman Salah, and Hesham M Eraqi. 2021. Asmdd: Arabic speech mispronunciation detection dataset. *arXiv preprint arXiv:2111.01136*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Fatma Zahra Besdouri, Inès Zribi, and Lamia Hadrich Belguith. 2024. Arabic automatic speech recognition: challenges and progress. *Speech Communication*, 163:103110.

Yassine El Kheir, Amit Meghanani, Hawau Olamide Toyin, Nada Almarwani, Omnia Ibrahim, Youssef Elshahawy, Mostafa Shahin, and Ahmed Ali. 2025. Iqra'eval: A shared task on qur'anic pronunciation assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*. Association for Computational Linguistics.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. 1993. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Nawar Halabi and Mike Wald. 2016. Phonetic inventory for an Arabic speech corpus. pages 734–738, Portorož, Slovenia. European Language Resources Association (ELRA).

Alissa M Harrison, Wai-Kit Lo, Xiaojun Qian, and Helen Meng. 2009. Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training. In *SLaTE*, pages 45–48.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Shao-Wei Fan Jiang, Bi-Cheng Yan, Tien-Hong Lo, Fu-An Chao, and Berlin Chen. 2021. Towards robust mispronunciation detection and diagnosis for l2 english learners with accent-modulating methods. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1065–1070. IEEE.

Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Olamide Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, Salima Mdhaffar, Thomas Hain, Yasser Hifny, Mostafa Shahin, and Ahmed Ali. 2025. Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study. *Preprint*, arXiv:2506.07722.

Wai-Kim Leung, Xunying Liu, and Helen Meng. 2019. Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8132–8136. IEEE.

Kun Li, Xiaojun Qian, and Helen Meng. 2016. Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):193–207.

Helen Meng, Yuen Yee Lo, Lan Wang, and Wing Yiu Lau. 2007. Deriving salient learners' mispronunciations from cross-language phonological comparisons. In *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 437–442. IEEE.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, and 1 others. 2022. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1179–1210.

Faria Nazir, Muhammad Nadeem Majeed, Mustansar Ali Ghazanfar, and Muazzam Maqsood. 2019. Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for arabic phonemes. *IEEE Access*, 7:52589–52608.

Ambra Neri, Ornella Mich, Matteo Gerosa, and Diego Giuliani. 2008. The effectiveness of computer assisted pronunciation training for foreign language learning by children. *Computer Assisted Language Learning*, 21(5):393–408.

Linkai Peng, Kaiqi Fu, Binghuai Lin, Dengfeng Ke, and Jinsong Zhang. 2021. A study on fine-tuning wav2vec2. 0 model for the task of mispronunciation detection and diagnosis. In *Interspeech*, volume 2021, pages 4448–4452.

Xiaojun Qian, Helen Meng, and Frank Soong. 2010. Capturing l2 segmental mispronunciations with joint-sequence models in computer-aided pronunciation training (capt). In *2010 7th International Symposium on Chinese Spoken Language Processing*, pages 84–88. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Silke M Witt and Steve J Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech communication*, 30(2-3):95–108.

Minglin Wu, Kun Li, Wai-Kim Leung, and Helen Meng. 2021. Transformer based end-to-end mispronunciation detection and diagnosis. In *Interspeech*, pages 3954–3958.

Shu-wen Yang, Heng-Jui Chang, Zili Huang, Andy T Liu, Cheng-I Lai, Haibin Wu, Jiatong Shi, Xuankai Chang, Hsiang-Sheng Tsai, Wen-Chin Huang, and 1 others. 2024. A large-scale evaluation of speech foundation models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2884–2899.

Nishmia Ziafat, Hafiz Farooq Ahmad, Iram Fatima, Muhammad Zia, Abdulaziz Alhumam, and Kashif Rajpoot. 2021. Correct pronunciation detection of the arabic alphabet using deep learning. *Applied Sciences*, 11(6):2508.

ANPLers at IqraEval Shared task: Adapting Whisper-large-v3 as Speech-to-Phoneme for Qur'anic Recitation Mispronunciation Detection

Nour Qandous[1], Serry Sibaee[2*], Samar Ahmed[1], Omer Nacar[3], Yasser Al-Habashi[2]
Adel Ammar[2], Wadii Boulila[2]
[1]NAMAA, Riyadh, Saudi Arabia
[2]Prince Sultan University, Riyadh, Saudi Arabia
[3]Tuwaiq Academy – Tuwaiq Research and Development Center, Riyadh, Saudi Arabia

Abstract

Mispronunciation detection at the phoneme level provides detailed feedback for Quranic reciters. Standard speech-to-text models cannot capture subtle differences in letter pronunciation; thus, developing a speech-to-phoneme system is essential. Prior works have mainly explored encoder-only models. In this work, we adapt Whisper-large-v3 on the IqraEval dataset. Our experimental results show that the proposed system achieved an F1-score of 0.3224, an accuracy of 0.6894, and a high recall of 0.7624. These results highlight promising directions for further research and development in phoneme-level mispronunciation detection.

1 Introduction

Computer-aided language Learning (CALL) employs computer technologies to aid in language acquisition and pronunciation. Accurate pronunciation is critical in the context of Arabic language learning, particularly for Quranic recitation, as it has both linguistic and religious significance.

Speech-to-Phoneme (STP) transcribe audio to tokens of phonemes; it differs from conventional Speech-to-Text (STT) systems that transcribe audio into tokens of words. Unlike STT, STP provides a finer-grained phonetic representation that is essential for precise pronunciation feedback and error detection, particularly relevant for Quranic recitation where subtle phonetic distinctions alter meaning and correctness.

Given the importance of phoneme-level accuracy for Quranic recitation, STP systems offer a promising approach to support learners in mastering Quranic pronunciation by analyzing their recitation audio record, detecting any fine-grained error, then provide a corrective feedback.

In this work, we fine-tuned **Whisper-large-v3** as an STP model to help detect mispronunciation of the Quranic recitation at the phoneme level. This paper is organized as follows: Section 2 reviews related works in mispronunciation detection using phonemes in Quranic recitation. Section 3 describes the proposed system and the experimental setup. Section 4 reports and analyzes the results. Section 5 discusses the results and provides insights. Finally, Section 6 concludes the paper and outlines possible directions for future work.

2 Related Works

Building on prior work in Quranic recitation recognition(Al-Zaro et al., 2025) developed a phoneme-based speech recognition system for Quranic recitation using the DeepSpeech architecture. They proposed a phoneme list of 53 units, covering consonants, vowels, and Tajweed-specific phonemes. The system was trained on a combined dataset consisting of a proprietary corpus from EqraTech and the open-source ASR Tarteel dataset, due to the lack of publicly available phoneme-level datasets for the Quran, totaling approximately 550 hours of audio. Evaluation results reported a phoneme error rate (PER) of 7.4%, a word phoneme error rate (WPER) of 27.97%, and a word error rate (WER) of 3.92% at the `imlā'ī` (spelling) word level.

Similarly(Calik et al., 2023) presented an ensemble-based framework for detecting mispronunciations of Arabic phonemes, particularly in the context of Quranic pronunciation, utilizing machine learning techniques including SVM, k-NN, and decision trees. The system also used feature extraction methods to enhance language learning through computer-assisted tools. It employed both traditional and ensemble learning-based approaches for evaluation. An accuracy of 95.9% was achieved using a voting classifier with mel-spectrogram features.

Extending the focus to Tajweed-specific errors(Harere and Jallad, 2023) developed an au-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 464–468
November 8-9, 2025 ©2025 Association for Computational Linguistics

tomatic mispronunciation detection system for Quranic recitation based on Tajweed rules (Separate Stretching, Tight Noon, and Hide), using the QDAT dataset, a public dataset containing over 1,500 audio samples of correct and incorrect recitations. The system addressed the shortage of qualified human supervisors by leveraging deep learning, specifically Long Short-Term Memory (LSTM) networks. The model achieved high accuracy rates of 96% for Separate Stretching, 95% for Hide, and 96% for Tight Noon. In a broader context of language learning (Algabri et al., 2022) applied deep learning to develop a versatile high-performance assisted pronunciation system for the detection, diagnosis, and generation of articulatory feedback for non-native Arabic learners. It used YOLO-based object detection for phoneme and articulatory feature recognition and employed a CNN-RNN-CTC model to provide feedback. The system achieved a phoneme error rate (PER) of 3.83% in the phoneme recognition task, an F1-score of 70.53% in the mispronunciation detection and diagnosis task, and a detection error rate (DER) of 2.6% in the articulatory feature detection task.

Most recently, (Şükrü Selim Çalık et al., 2024) introduced a novel framework for mispronunciation detection of Arabic phonemes using audio-based transformer models such as Squeezed and Efficient Wav2Vec (SEW), Hidden-Unit BERT (HUBERT), Wav2Vec, and UNI-SPEECH. The study is considered the first to comprehensively explore Arabic phoneme mispronunciations using these models. A dataset consisting of 29 Arabic phonemes, including 8 hafiz sounds, was collected from 11 speakers and supplemented with additional samples from YouTube. The results demonstrated that the UNI-SPEECH model achieved superior performance. Moreover, the proposed framework was designed to be speaker-independent, allowing for general applicability without the need for individual speaker enrollment.

Previous studies show us that they focused on using encoder-only architecture in solving such a problem. Therefore, we are exploring a new direction by investigating encoder–decoder architecture, namely the Whisper model, to detect mispronunciation on phoneme level.

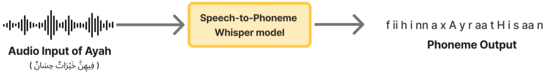

Audio Input of Ayah
(فِيهِنَّ خَيْرَاتٌ حِسَانٌ)
Speech-to-Phoneme Whisper model
f ii h i nn a x A y r aa t H i s aa n
Phoneme Output

Figure 1: Overview of the proposed speech-to-phoneme system.

3 Methodology

3.1 Dataset

We utilized the complete training and evaluation subsets from the IqraEval dataset (El Kheir et al., 2025) for model training, yielding a total of 73,990 records and approximately 82.4 hours of audio. The dataset consists of short Arabic audio recorded by multiple speakers, paired with the corresponding sentences and detailed phoneme transcriptions. Each sample also includes metadata such as a unique identifier and the sentence with tashkeel. Although the dataset contains multiple attributes, we only utilized the audio and phoneme attributes for the speech-to-phoneme (STP) task.

3.2 Proposed System

Figure 1 presents the proposed speech-to-phoneme (STP) system, in which an audio recording of recitation is transcribed into a phoneme sequence using our STP model. We base our system on **Whisper-large-v3**, the largest Whisper variant with 1550M parameters and multilingual support. The reason for selecting **Whisper-large-v3** is that it achieved outstanding performance on the Arabic Automatic Speech Recognition (ASR) leaderboard (Wang et al., 2024) with an average word error rate of 0.3686, and because it is faster to fine-tune compared to other exceptional models. Our assumption was that if an STT model achieved high performance on processing Arabic audio into words, it would get promising performance in transcribing Arabic Audio into phonemes, and we will discuss this assumption later in the discussion section 5.

To enable phoneme-level prediction, we extend the original Whisper tokenizer by incorporating 68 additional phoneme tokens from (Halabi and Wald, 2016), and resize the embedding layer to match the new vocabulary. This modification yields an adapted Whisper model capable of generating phoneme sequences directly from audio inputs.

Breaking down the system as illustrated in Figure 2, the Whisper feature extractor first computes log-mel spectrograms from the input audio. These

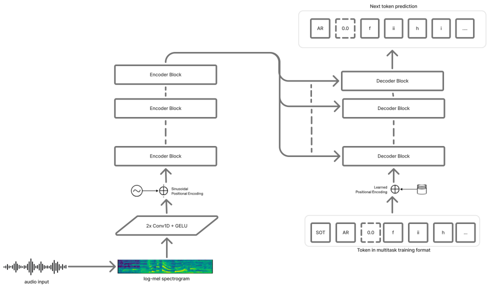


Figure 2: Whisper architecture for the speech-to-phoneme task, using a custom tokenizer with 68 phoneme tokens.

spectrograms are processed by the encoder to produce high-level acoustic representations. The decoder then predicts the output sequence token-by-token, conditioned on the preceding phoneme tokens, until the complete phoneme transcription is generated. During training, phoneme sequences are used as target labels to guide the prediction process.

3.3 Hyperparameter Optimization

This training setup optimizes for efficient fine-tuning by using a small batch size with gradient accumulation to simulate a larger effective batch, a low learning rate for stable updates, and mixed precision (fp16) to reduce memory usage. It saves checkpoints frequently while limiting total saved models, evaluates performance by word error rate (WER) after each epoch, and aims to minimize WER for best model selection. The training was run on 3 A100 80GB GPUs, leveraging a batch size of 4 per device and gradient accumulation over 3 steps to optimize memory usage and training efficiency.

4 Results

$$Precision = \frac{TR}{TR + FR} \tag{1}$$

$$Recall = \frac{TR}{TR + FA} \tag{2}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

Table 1 presents the evaluation metrics of our system on the Iqraeval test set. True Rejects (TR) are the percentage of mispronunciations correctly transcribed as phonemes, while False Accepts (FA) represent the percentage of failures to transcribe mispronunciations. True Accepts (TA) demonstrate the correct transcription of correct pronunciation, while False Rejects (FR) are the percentage of incorrect transcription of correct pronunciation. See Equations 1, 2, and 3 for more context.

As shown in Table 1, the model demonstrated a high recall of 0.7624 and a correct rate of 0.7682, indicating strong capability in transcribing mispronounced phonemes and covering phoneme variations effectively. However, the relatively low precision of 0.2045 suggest that the system suffers from a high number of False Rejects (FR), yielding a low F1-score of 0.3224.

The true acceptance rate (TA) of 0.7868 shows that the majority of correctly pronounced phonemes are transcribed correctly, yet the False Accept (FA) of 0.2376 reveals that a significant portion of mispronunciations remains undetected. Accuracy and correct detection rate (CD), at 0.6894 and 0.5418, respectively, indicate moderate overall classification quality.

These results demonstrate that our system can be optimized to transcribe fine-grained features, and there is room to improve the precision and reduce the false rejects to achieve a more balanced and effective phoneme recognition performance.

Metric	Value
F1-score	0.3224
Precision	0.2045
Recall	0.7624
Correct Rate	0.7682
Accuracy	0.6894
TA	0.7868
FR	0.2132
FA	0.2376
CD	0.5418

Table 1: Test set results for the proposed system.

5 Discussion

Table 2 shows model predictions for three recordings of the same Ayah [78-Al Imran] with supposedly different pronunciations. Although the model should generate three different transcriptions for these three recordings, it generates identical transcriptions for the first two and slightly different transcriptions for the third one. As illustrated in bold in Table 2, the difference was in a **single phoneme** in the word **yalwun**. All three recordings

ID	Prediction
00000_00013	w a n m i n h u m l a f a r ii q AA y a l w u n **u** E a l s i n a t a h u m b i l k i t aa b
00000_00013143	w a n m i n h u m l a f a r ii q AA y a l w u n **u** E a l s i n a t a h u m b i l k i t aa b
00000_00013343	w a n m i n h u m l a f a r ii q AA y a l w u n **a** E a l s i n a t a h u m b i l k i t aa b

Table 2: Outputs of the proposed system for three recordings identified by ID metadata from the IqraEval test set, all corresponding to Ayah 78 of Surah Al-Imran.

could appear identical to a regular Arabic listener, except that the end of the word **yalwun** needs to be clearer from the speaker, or, as said in Tajweed, the sound needs more duration. So from our perspective, it is difficult for an Arabic listener to predict the correct phoneme.

Moreover, prior works demonstrated that using CTC-based models with self-supervised encoders such as Wav2Vec 2.0 achieved high performance in phoneme-level mispronunciation detection (Kheir et al., 2025). Our approach adopts an encoder–decoder paradigm with **Whisper-large-v3**. The experimental results show the effectiveness of our approach. The downside is the adaptation limitations that require significant hardware and prevent real-time operation, as the model has 1500M parameters. Quantizing the model or using smaller versions and increasing the number of epochs could provide more robust performance and a more reliable system.

5.1 Analysis of the dataset

Our comprehensive analysis of the Arabic voice dataset revealed several critical quality issues that warrant careful consideration for the shared task implementation. Technical artifacts were prevalent throughout the collection, with numerous audio samples exhibiting signal truncation and cutting issues that compromise the integrity of the speech data. A substantial portion of recordings demonstrated incorrect application of tahreek (diacritical markings) at sentence endpoints, deviating from standard Arabic phonological conventions for proper vocalization. While the dataset predominantly consists of Quranic recitations, we identified instances of mispronunciation that diverge from canonical tajweed principles, potentially introducing phonetic inconsistencies in model training. Temporal irregularities were observed across the corpus, with speaking rates varying significantly from normal conversational pace, which may adversely affect automatic speech recognition performance and temporal alignment algorithms. Furthermore, we detected grammatical errors within the spoken content and critical misalignments between reference transcriptions and their corresponding audio files, representing fundamental data integrity challenges. These systematic quality control issues necessitate robust preprocessing pipelines and filtering mechanisms to ensure dataset reliability and maintain the validity of experimental results in Arabic speech processing applications.

6 Conclusion

Mispronunciation detection is a challenging task, and it becomes even harder in Quranic recitation scenarios; thus, phoneme-level mispronunciation detection is essential. In this paper, we explored the use of the Whisper model and addressed this problem as a speech-to-phoneme task. Our results reveal a substantial gap between a recall of 0.7624 and a precision of 0.2045, indicating that while the model effectively identifies a wide range of mispronounced phonemes, it frequently misclassifies correctly pronounced ones. A limitation of this work results from the hardware requirements that prevented us from experimenting with more epochs. For future work, we encourage researchers to explore encoder–decoder architectures for phoneme-level mispronunciation detection, investigate Nvidia Conformer CTC Arabic models, which combine the Conformer architecture with CTC and have shown high Arabic STT performance (Wang et al., 2024). These directions could further advance research in this area, provide a more reliable system to improve Quranic recitations for Muslims, and inspire new ideas for mispronunciation detection.

References

Samah Al-Zaro, Mahmoud Al-Ayyoub, and Osama Al-Khaleel. 2025. Speaker-independent phoneme-based automatic quranic speech recognition using deep learning. *IEEE Access*, 13:125881–125896.

Mohammed Algabri, Hassan Mathkour, Mansour Alsulaiman, and Mohamed A Bencherif. 2022. Mispronunciation detection and diagnosis with articulatory-level feedback generation for non-native arabic speech. *Mathematics*, 10(15):2727.

Sukru Selim Calik, Ayhan Kucukmanisa, and Zeynep Hilal Kilimci. 2023. An ensemble-based framework for mispronunciation detection of arabic phonemes. *Preprint*, arXiv:2301.01378.

Yassine El Kheir, Amit Meghanani, Hawau Olamide Toyin, Nada Almarwani, Omnia Ibrahim, Youssef Elshahawy, Mostafa Shahin, and Ahmed Ali. 2025. Iqra'eval: A shared task on qur'anic pronunciation assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*. Association for Computational Linguistics.

Nawar Halabi and Mike Wald. 2016. Phonetic inventory for an Arabic speech corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 734–738, Portorož, Slovenia. European Language Resources Association (ELRA).

Ahmad Al Harere and Khloud Al Jallad. 2023. Mispronunciation detection of basic quranic recitation rules using deep learning. *Preprint*, arXiv:2305.06429.

Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Olamide Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, Salima Mdhaffar, Thomas Hain, Yasser Hifny, Mostafa Shahin, and Ahmed Ali. 2025. Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study. *Preprint*, arXiv:2506.07722.

Yingzhi Wang, Anas Alhmoud, and Muhammad Alqurishi. 2024. Open universal arabic asr leaderboard. *arXiv preprint arXiv:2412.13788*.

Şükrü Selim Çalık, Ayhan Küçükmanisa, and Zeynep Hilal Kilimci. 2024. A novel framework for mispronunciation detection of arabic phonemes using audio-oriented transformer models. *Applied Acoustics*, 215:109711.

AraS2P: Arabic Speech-to-Phonemes System

Bassam Mattar
Alexandria University
b.mattar@alexu.edu.eg

Mohamed Fayed
Applied innovation Center
m.essam@aic.gov.eg
Georgia Institute of Technology
mfayed8@gatech.edu

Ayman Khalafallah
Applied innovation Center
a.khalafallah@aic.gov.eg

Abstract

This paper describes AraS2P, our speech-to-phonemes system submitted to the Iqra'Eval 2025 Shared Task. We adapted Wav2Vec2-BERT via Two-Stage training strategy. In the first stage, task-adaptive continue pretraining was performed on large-scale Arabic speech-phonemes datasets, which were generated by converting the Arabic text using the MSA Phonetiser. In the second stage, the model was fine-tuned on the official shared task data, with additional augmentation from XTTS-v2-synthesized recitations featuring varied Ayat segments, speaker embeddings, and textual perturbations to simulate possible human errors. The system ranked first on the official leaderboard, demonstrating that phoneme-aware pretraining combined with targeted augmentation yields strong performance in phoneme-level mispronunciation detection.

1 Introduction

Automatic mispronunciation detection and diagnosis (MDD) plays a key role in computer-aided pronunciation learning (CAPL), providing learners with objective and scalable feedback on their pronunciation quality score (Kheir et al., 2023). In Arabic, MDD is particularly challenging due to the language's complex phonemic inventory, the presence of emphatic and pharyngeal consonants, and the semantic role of short vowels (diacritics) (Abdou and Rashwan, 2014). These characteristics make accurate phoneme-level detection especially important, as even subtle deviations can significantly change meaning.

In this work, we describe a system based on a Wav2Vec2-BERT architecture (Baevski et al., 2020) that employs a two-stage training strategy: (1) task-adaptive continue pretraining on large Arabic speech datasets—Common Voice (Arabic split), SADA, and MASC—using phoneme-level supervision generated via the MSA Phonetiser,[1] resulting in labeled corpora that capture fine-grained phonetic distinctions, and (2) fine-tuning on the official shared task data as well as targeted augmentation through XTTS-v2-synthesized recitations that vary in Ayat segments, speaker embeddings, and noisy textual content to simulate realistic recitations errors.

We summarize our contributions as follows:

- A phoneme-aware task-adaptive pretraining strategy for Arabic MDD using large-scale speech-phonemes data.
- A targeted augmentation pipeline where we add noise to text, convert the noisy text to phonemes using MSA-Phonetizer, and generate corresponding speech for many speakers using XTTS-v2 (Casanova et al., 2024).
- Our model ranks first on the Iqra'Eval 2025 benchmark leaderboard, demonstrating effectiveness of our training strategy.

2 Related Work

2.1 Arabic CAPL and Mispronunciation Detection

Computer-Assisted Pronunciation Learning (CAPL) systems rely on Mispronunciation Detection and Diagnosis (MDD) to provide automated feedback for learners (Witt and Young, 2000; Eskenazi, 2009). Early MDD approaches often derived pronunciation quality metrics from acoustic likelihoods computed from recognition results, such as the Goodness of Pronunciation (GOP) score (Witt and Young, 2000). While GOP provides a practical way to detect pronunciation deviations, its granularity is limited to the phone level and its accuracy can be affected by recognition errors. Other research (Bonaventura

[1] https://github.com/Iqra-Eval/MSA_phonetiser

et al., 2000; Raux and Kawahara, 2002) has enhanced pronunciation modeling by incorporating likely pronunciation variants into a pronunciation dictionary, which can involve manual specification of error patterns.

Recent years have seen the adoption of deep learning and end-to-end architectures for MDD, enabling systems to learn pronunciation error patterns directly from data (Peng et al., 2022). For Arabic, MDD poses additional challenges due to its rich consonant inventory, emphatic and pharyngeal sounds, and the omission of short vowels in most written text and ASR systems (Kheir et al., 2025). Consequently, slight pronunciation errors—such as mixing up emphatic and non-emphatic consonants—may change the meaning of a word.

Arabic MDD research has explored handcrafted acoustic features, CNN-based classifiers, and transfer learning from large-scale ASR models (Calık et al., 2023; Alrashoudi et al., 2025). Several works have focused on Qur'anic recitation, where precise phoneme articulation is central (Abdou and Rashwan, 2014; Alrumiah and Al-Shargabi, 2023; Harere and Jallad, 2023). (Kheir et al., 2025) provided the first publicly available benchmark for Arabic phoneme-level MDD, using Qur'anic recitation with time-aligned phoneme annotations.

2.2 Self-Supervised Phoneme Recognition Models

Self-supervised learning has significantly advanced phoneme recognition, which in turn has improved the performance of MDD systems. Wav2Vec-BERT 2.0 model (Baevski et al., 2020) learns contextualized speech representations from raw audio by combining a convolutional encoder with a Transformer context network (Devlin et al., 2019; Baevski et al., 2019). It was pretrained using a contrastive objective (Chen et al., 2020; He et al., 2020) over masked audio segments, then fine-tuned with a Connectionist Temporal Classification (CTC) objective (Graves et al., 2006). Wav2vec 2.0 achieves state-of-the-art performance in phoneme recognition tasks, making it well-suited for MDD.

Building on this, Wav2Vec-BERT integrates a BERT-style masked language modeling (MLM) objective (Devlin et al., 2019) with the Wav2Vec 2.0 framework (Chung et al., 2021). This joint optimization learns both quantized acoustic units and contextual relationships between them, producing richer and more discriminative phonetic representations. Instead of iteratively re-clustering discrete units like HuBERT (Hsu et al., 2021), w2v-BERT learns quantization and context modeling in a single end-to-end process.

Multilingual Wav2Vec-BERT 2.0 extends this approach to 143 languages using over 4.5 million hours of speech for pretraining (Barrault et al., 2023). Its large-scale multilingual exposure enables robust representation of fine phonetic distinctions, even in low-resource settings like Arabic MDD. Compared to Wav2Vec 2.0, Wav2Vec-BERT 2.0 incorporates MLM-based contextual modeling directly into the acoustic encoder, allowing it to learn longer-range phoneme patterns. For this reason, we used Wav2Vec-BERT 2.0 pretrained weights.

2.3 Benchmarks and Shared Tasks

Iqra'Eval Shared Task (Kheir et al., 2025) represents a milestone for Arabic MDD by offering a publicly available benchmark, standardized evaluation protocol, and a leaderboard for reproducible comparison. Similar to MGB Challenge for Arabic ASR (Ali et al., 2016) and other shared tasks in speech and Natural Language Processing (NLP), this benchmark has stimulated community engagement and methodological innovation. Through integrating controlled evaluation with phoneme-level detection, Iqra'Eval addresses a critical gap in Arabic CAPL research by establishing a standardized benchmark for systematic evaluation.

3 Two-Stage Training

We adapted Wav2Vec2-BERT (Barrault et al., 2023) to our downstream task via Two-Stage training. We continued pretraining it on Arabic speech-phonems pairs (section 3.1). Meanwhile, we conducted exploratory data analysis to measure the alignment between continue pretraining and fine-tuning (section 3.2). Finally, we utilized training set of the task as well as our synthetically generated dataset for fine-tuning (section 3.3).

3.1 Adaptive Continue Pretraining

Continue pretraining has shown to be an effective technique to improve the performance of pretrained models on languages of interest (Kalyan et al., 2021; Zhou et al., 2024; Fujii et al., 2024; Alves et al., 2024). To boost our model, we continued pretraining it on speech-phonemes pairs. We deployed MSA-phonetizer[2] to convert open-

[2] https://github.com/Iqra-Eval/MSA_phonetiser

source datasets with speech-text pairs into speech-phonemes pairs, hence adapting it to suite the downstream task (Adaptive Continue Pretraining). Specifically, our pretraining data is constructed from Common Voice Arabic split (Ardila et al., 2019), SADA(Alharbi et al., 2024) and MASC(Al-Fetyani et al., 2023) datasets. Table1 includes statistics about these datasets.

Dataset	size (hours)
Common Voice (Ar-Split)	157
SADA	668
MASC	1,000

Table 1: Statistics of datasets used in our adaptive continue pretraining stage.

We used Adam optimizer with weight decay (Loshchilov and Hutter, 2017). We set hyper-parameters as follows: learning rate of 1×10^{-5}, Linear Decay scheduler, weight decay equals to 0.01, Adam betas of $(0.9, 0.999)$, gradient clipping at 1.0, and batch size of 32. We continue the pretraining for $800k$ iterations.

3.2 Exploratory Data Analysis

We have had a hypothesis that there is a discrepancy between pretraining data and fine-tuning one. So, we plotted the histogram of the most frequent phonemes in both the pretraining and training datasets. As shown in figure 1, the distributions of phonemes differ notably, particularly for elongated phonemes such as "aa," "ii," "uu," and "AA.". This observation confirms the correctness of our hypothesis and highlights the importance of further fine-tuning on downstream task.

Prior to fine-tuning, we notice a difference between the phoneme inventory in the training dataset and the phonemes produced by the MSA phonetizer. We align the phonemes as shown in Table 2.

3.3 Fine-tuning

After continuing pretraining, we performed vanilla fine-tuning for the model on our "Tuning dataset" 3.3.1. We used the same training parameters as that of continue pretraining.

3.3.1 Tuning dataset

To further align the model with the task, we used the training set provided with the task, and created synthetic dataset to increase overall data size. Preparing the synthetic data has went through two main stages: prepare the noisy text and generate corresponding audio files.

Phonetiser Phoneme	Inventory Phoneme
II0	II
I0I0	II
I0	I
I1	I
ii0	ii
i0i0	ii
i0	i
i1	i
UU0	UU
U0	U
U1	U
uu0	uu
u0u0	uu
uu1	uu
u0	u
u1	u

Table 2: Mapping from MSA phonetizer output to the training dataset phoneme inventory.

Prepare Noisy Text: We downloaded the text of the holy quran and perturbed the text with what we consider to be valid noise. The algorithm to generate valid noise is shown in algorithm1.

Algorithm 1 Noising Algorithm

```
procedure GENERATENOISYTEXT(text, arabic_chars, noise_map, max_noise)
    $target\_noise \leftarrow RandInt(1, max\_noise)$
    $new \leftarrow$ empty list; $count \leftarrow 0$
    for ch in text do
        if $count >= target\_noise$ then
            Append ch
        else if $UniformRandom(0, 1) < p_{noise}$ then
            $count \leftarrow count + 1$
            Choose noise type: delete / substitute / insert
            if substitute then
                Append RandChoice (noise_map[ch])
            else if insert then
                Append RandChoice(arabic_chars), ch
            end if
        else
            Append ch
        end if
    end for
    return Join($new$)
end procedure
```

Audio Generation: We downloaded many audio files for various speakers to ensure the variety of data and to avoid overfiting over small set of speakers. Then, we generated speaker embeddings using embedder module in XTTS-v2 (Casanova et al.,

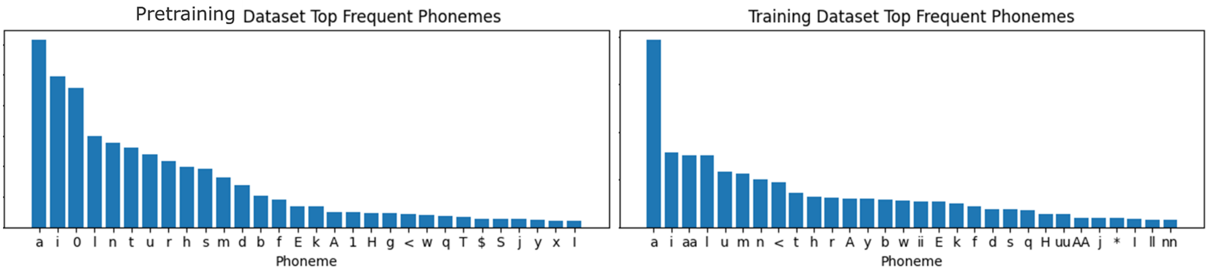


Figure 1: Histogram of top frequent phonemes in pseudo-labelled pretraining and training datasets

2024). Finally, we converted the noisy text to audio files using XTTS-v2.

The resulted dataset is 60 hours of audio files, and represented 30% of Tuning data.

While selecting checkpoint for testing, we noticed a shift in distribution between our valid set and competition's test set. Hence, we selected checkpoint saved after 2.5 epochs for submission to balance generalizability and good performance on the downstream task.

4 Results

In this section, we illustrate the metrics used (section 4.1), report quantitative results (section 4.2), and shows some examples from our qualitative analysis (section 4.3).

4.1 Metrics

The system is evaluated using several complementary metrics. First, the **Correct Rate** measures the proportion of phonemes that are detected correctly, and is defined as $1 -$ Phoneme Error Rate (PER). In addition, **Accuracy** captures the proportion of phonemes classified correctly as either pronounced correctly or mispronounced. To further distinguish system behavior, **True Acceptance (TA)** refers to cases where a correct phoneme is correctly accepted, while **True Rejection (TR)** corresponds to mispronounced phonemes that are correctly flagged. Conversely, errors are represented by **False Acceptance (FA)**, when a mispronunciation is missed, and **False Rejection (FR)**, when a correct phoneme is wrongly flagged. Beyond detection, **Correct Diagnosis (CD)** evaluates how often the system not only detects a mispronunciation but also identifies the specific mispronounced phoneme. Finally, the system's classification quality is summarized through **Precision**, defined as $\frac{TR}{TR+FR}$, **Recall**, defined as $\frac{TR}{TR+FA}$, and their harmonic mean, the **F1-score**, computed as $\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$.

4.2 Quantative Analysis

Table 3 shows the results of our system under different setups: after adaptive continue pretraining, fine-tuning on the official training data of the task, and after fine-tuning on our Tuning data. The results demonstrate that fine-tuning is essential for optimizing the system's alignment with Qur'anic recitation assessment. More importantly, they show the effectiveness of our synthetic data generation pipeline, achieving top performance across all of our systems.

4.3 Qualitative Analysis

Table 4 presents examples from both the fine-tuning on training set only setup and the continued pretraining-one. Because of time constraints and high similarity between fine-tuning on training set only and on Tuning set, we leave its qualitative analysis for future work. The results indicate that the system trained with pretraining alone fails to accurately predict phonemes associated with diacritics, particularly the "shadda". This limitation is likely due to the rarity of such phonemes in the pretraining data as discussed in subsection 3.1. This further confirms that adaptive continue pretraining was not sufficient and that we need for fine-tuning on the training set of the task.

5 Conclusion

In this work, we illustrated our recipe to adapt Wav2Vec-BERT 2.0 on speech-to-phoneme task. First, adaptively continued pretraining it on Arabic speech-phonemes corpora. Second, we prepared synthetic data for fine-tuning phase by generating noisy text, convert it to phonemes using MSA-phonetizer, and generate corresponding speech for many speakers using XTTS-v2. Our model scored first on IqraEval 2025, illustrating the ffectiveness of our approach.

System	F1↑	Prec.↑	Rec.↑	CR↑	Acc.↑	TA↑	FR↓	FA↓	CD↑
pretraining only	0.1923	0.1091	0.807	0.5156	0.5117	0.5264	0.4736	**0.193**	0.4363
fine-tuning									
training data	0.4561	0.3327	0.7252	0.8714	0.8576	0.8954	0.1046	0.2748	0.568
Tuning data	**0.4726**	0.3713	0.6501	**0.8985**	**0.8701**	**0.9209**	**0.0791**	0.3499	**0.6873**

Table 3: Performance on the Iqra'Eval 2025 leaderboard. CR = Correct Rate, Acc. = Accuracy, TA = True Acceptance, FR = False Rejection, FA = False Acceptance, CD = Correct Diagnosis.

Ref. Aya Segment	Recited Aya Segment (With Error)	Pretrained System Output	Fine-tuned System Output
يُغَشِّيكُمُ النُّعَاسَ أَمَنَةً مِنْهُ	يُخَشِّيكُمُ النُّعَاسُ أَمَنَةً مِنْهُ	ii x $ ii k m l n E aa s m n h m n h	y u x A $$ ii k u m u l nn u E aa s u < a m a n a t a n mm i n h u
إِيَّاكَ نَعْبُدُ	إِيَّاكَ نَعْبَدُ	y aa k n E b d	< ii y aa k a n a E b a d u
ذَلِكَ الْكِتَابُ لَا رَيْبَ فِيهِ	ذَلِكَ الْكِتَابُ لَا رَيْبُ فِيهِ	* aa l i k l k t aa b l aa r ii b f ii h	* aa l i k a l k i t a b u l aa r a y b a f ii h i
ٱلرَّحمَنَ	ٱلرَّرَّرَّحمَنَ	a l r H m n	l rr rr rr a H m a n i

Table 4: Comparison Between Only Pretrained and Fine-tuned System

6 Acknowledgment

We thank the IqraEval organizers(El Kheir et al., 2025) for their support and for providing clear documentation and tools such as the evaluation API. We also acknowledge the contributors of the open-source datasets we used. This effort was supported by Applied Innovation Center (AIC) 's High Performance Computing facilities

References

Sherif Mahdy Abdou and Mohsen Rashwan. 2014. A computer aided pronunciation learning system for teaching the holy quran recitation rules. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pages 543–550. IEEE.

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith Abandah, Adham Alsharkawi, and Maha Dawas. 2023. Masc: Massive arabic speech corpus. In *IEEE-SLT*.

Sadeen Alharbi, Areeb Alowisheq, Zoltán Tüske, Kareem Darwish, Abdullah Alrajeh, Abdulmajeed Alrowithi, Aljawharah Bin Tamran, Asma Ibrahim, Raghad Aloraini, Raneem Alnajim, and 1 others. 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.

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. pages 279–284.

Norah Alrashoudi, Hend Al-Khalifa, and Yousef Alotaibi. 2025. Improving mispronunciation detection and diagnosis for non-native learners of the arabic language. *Discover Computing*, 28(1):1.

Sarah S Alrumiah and Amal A Al-Shargabi. 2023. Intelligent quran recitation recognition and verification: Research trends and open issues. *Arabian Journal for Science and Engineering*, 48(8):9859–9885.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, and 1 others. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2019. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, and 1 others. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.

Patrizia Bonaventura, Daniel Herron, and Wolfgang Menzel. 2000. Phonetic rules for diagnosis of pronunciation errors. In *KONVENS 2000/Sprachkommunikation, Vorträge der gemeinsamen Veranstaltung 5. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS), 6. ITG-Fachtagung" Sprachkommunikation"*, pages 225–230.

Sükrü Selim Calık, Ayhan Kucukmanisa, and Zeynep Hilal Kilimci. 2023. An ensemble-based framework for mispronunciation detection of arabic phonemes. *Applied Acoustics*, 212:109593.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and 1 others. 2024. Xtts: a massively multilingual zero-shot text-to-speech model. *arXiv preprint arXiv:2406.04904*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.

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.

Yassine El Kheir, Amit Meghanani, Hawau Olamide Toyin, Nada Almarwani, Omnia Ibrahim, Youssef Elshahawy, Mostafa Shahin, and Ahmed Ali. 2025. Iqra'eval: A shared task on qur'anic pronunciation assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*. Association for Computational Linguistics.

Maxine Eskenazi. 2009. An overview of spoken language technology for education. *Speech communication*, 51(10):832–844.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. *arXiv preprint arXiv:2404.17790*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Ahmad Al Harere and Khloud Al Jallad. 2023. Quran recitation recognition using end-to-end deep learning. *arXiv preprint arXiv:2305.07034*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.

Yassine El Kheir, Ahmed Ali, and Shammur Absar Chowdhury. 2023. Automatic pronunciation assessment–a review. *arXiv preprint arXiv:2310.13974*.

Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Olamide Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, and 1 others. 2025. Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study. *arXiv preprint arXiv:2506.07722*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Linkai Peng, Yingming Gao, Binghuai Lin, Dengfeng Ke, Yanlu Xie, and Jinsong Zhang. 2022. Text-aware end-to-end mispronunciation detection and diagnosis. *arXiv preprint arXiv:2206.07289*.

Antoine Raux and Tatsuya Kawahara. 2002. Automatic intelligibility assessment and diagnosis of critical pronunciation errors for computer-assisted pronunciation learning. In *INTERSPEECH*, pages 737–740.

Silke M Witt and Steve J Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech communication*, 30(2-3):95–108.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. 2024. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*.

Metapseud at Iqra’Eval: Domain Adaptation with Multi-Stage Fine-Tuning for Phoneme-Level Qur’anic Mispronunciation Detection

Ayman Mansour
Independent Researcher
aymanmnsor777@gmail.com

Abstract

This paper presents the Metapseud system designed for the Iqra’Eval shared task, which addresses the automatic assessment of Qur’anic recitation pronunciation, as part of ARABIC-NLP 2025. This system applies multi-stage fine-tuning of Wav2Vec2.0 with curriculum-inspired training, followed by domain adaptation to Qur’anic phoneme annotations. The decoding is improved using beam search with a CTC-based decoder. The results show that staged adaptation achieved a phoneme error rate (PER) of 21% in the development set, and beam search improves the accuracy in the open test set from 76.9% to 82.1%. The findings of this work emphasize the significance of curriculum learning, domain adaptation, and decoding strategies in recognizing mispronunciation in Qur’anic recitation.

1 Introduction

Mispronunciation Detection and Diagnosis (MDD) forms the core of modern Computer-Aided Pronunciation Training (CAPT) systems, providing real-time identification and analysis of learner pronunciation errors. By combining automated speech recognition (ASR), phonetics-driven error detection, and adaptive feedback mechanisms, MDD enables CAPT systems to not only assess pronunciation accuracy but also deliver targeted, pedagogically informed corrective guidance (Neri et al., 2008).

Qur’anic Arabic Automatic Speech Recognition (ASR) presents unique challenges due to its rich phonetic variation, complex Tajweed rules, and significant differences from Modern Standard Arabic (MSA) or dialectal Arabic—placing it in a distinct category often regarded as Classical Arabic (CA) (Habash, 2010). The accurate recitation of the Holy Quran is of profound importance to Muslims worldwide, as it must adhere to precise pronunciation rules (Tajweed), where even minor deviations can alter the meaning entirely. This necessity has motivated initiatives such as the Iqra’Eval shared task (El Kheir et al., 2025), which challenges researchers to develop automatic phoneme-level recognition systems for Qur’anic recitation.

The Metapseud system, which leverages self-supervised ASR model Wav2vec2.0 and combines (1) multistage curriculum-inspired fine-tuning, (2) domain adaptation, and (3) beam search decoding. This strategy is motivated by previous work on curriculum learning and domain adaptation for ASR.

2 Methodology

The methodology focuses primarily on applying Multi-stage fine-tuning by strategically leveraging the self-supervised learning capabilities of Wav2Vec 2.0 (Baevski et al., 2020) by integrating three core techniques: (1) **multistage curriculum-inspired fine-tuning**, (2) **targeted domain adaptation**, and (3) optimized **beam search decoding**. This integrated strategy is motivated by established principles in curriculum learning (Bengio et al., 2009; Platanios et al., 2019) and domain adaptation for speech (Kunze and et al., 2017; Wang and et al., 2021), applying them systematically to a single MDD pipeline for Qur’anic recitation.

2.1 Model Architecture and Foundation

The foundation of this system is the wav2vec2-large-xlsr-53-arabic (Grosman, 2021), based on wav2vec2-large-xlsr-53 a large model pre-trained in 53 languages (Conneau et al., 2020). This model is fine-tuned in Arabic using the train and validation splits of Common Voice 6.1 and Arabic Speech Corpus (Halabi, 2016), while it merely focuses on Arabic ASR, but it provides a robust starting point that subsequently specialize for the

Proceedings of The Third Arabic Natural Language Processing Conference, pages 475–479
November 8-9, 2025 ©2025 Association for Computational Linguistics

target domain.

2.2 Stage-1: General Domain Fine-Tuning (Curriculum Learning)

This stage first exposes the model to Qur'anic recitations to capture prosody and phoneme distributions. Used curriculum-inspired training by first training on a broader Qur'anic dataset that teaches the model general recitation structure and phoneme patterns, by gradually shifting from a general-purpose ASR model to a phoneme-centric Qur'anic recitation model. `Tarteel-ai-EA-DI` dataset ($\sim$245k) is a large and diverse corpus of Qur'anic recitations from various reciters (qurra'). This dataset prioritizes breadth to capture the full range of recitation styles and phonetic variations. The model is fine-tuned on this dataset using a standard Connectionist Temporal Classification (CTC) loss function with a phoneme-level vocabulary.

2.3 Stage-2: Domain Adaptation Fine-Tuning

This stage represents the final step in the curriculum, transitioning the model from a broad understanding to a specialized one. It directly implements domain adaptation (Kunze and et al., 2017; Wang and et al., 2021). To specialize the model for Qur'anic phoneme structures, was fine-tuned the previous model on the provided dataset `Iqra_train` (79 hours) of MSA speech augmented with Qur'anic recitations of Qur'anic phoneme-annotated recitations. This domain adaptation step aligns the model with Qur'anic-specific phoneme distributions, sharpening the model's focus on the specific acoustic features and phonological rules critical for accurate pronunciation evaluation, reducing PER significantly.

2.4 Beam Search Decoding

Beam search decoding is employed using the `PyCTCDecode`[1] library to generate the final phoneme sequences. This method improves upon greedy decoding (Graves et al., 2006; Hannun, 2017) and was chosen to find a more globally optimal sequence compared to the locally optimal stepwise predictions of greedy decoding. The decoding process was implemented using a standard `BeamSearchDecoderCTC` class, initialized with a phoneme vocabulary (`Iqra_train`) specific to this task.

2.5 Evaluation Metrics

The Model performance was evaluated using the hierarchical framework as (Kheir et al., 2023) which assesses both the detection and diagnosis of pronunciation errors, categorizing each predicted phoneme into one of several classes:

- True Accept (**TA**): A correct phoneme is correctly accepted.
- True Reject (**TR**): A mispronounced phoneme is correctly detected as an error.
- False Accept (**FA**): A mispronounced phoneme is incorrectly accepted (i.e., a missed error).
- False Reject (**FR**): A correct phoneme is incorrectly flagged as an error (i.e., a false alarm).

And Diagnosis-Level Categories, Correct Diagnosis (**CD**), Error Diagnosis (**ED**). From these categories, standard information retrieval metrics: **Precision**, **Recall**, and **F-measure** which derived from diagnostic accuracy, and widely used as the performance measures for mispronunciation detection.

$$Precision = \frac{TR}{TR + FR} \tag{1}$$

$$Recall = \frac{TR}{TR + FA} \tag{2}$$

$$F - measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

3 Data

3.1 Training and Development Data

3.1.1 Every Ayah Diacritized (EA-DI) dataset

The first stage of curriculum learning approach utilized the Every Ayah Diacritized (EA-DI) Phonemized dataset[2], a large-scale, based on Tarteel-Ai's Every Ayah Diacritized (EA-DI) dataset publicly available corpus It encompasses a wide variety of Qur'anic recitations, covering the entire text of the Qur'an from numerous reciters (qurra'). This diversity is crucial for teaching the model the broad acoustic properties and

[1] https://github.com/kensho-technologies/pyctcdecode

[2] https://huggingface.co/datasets/AymanMansour/tarteel-ai-EA-DI-phonemized-Final

phoneme distributions of the domain. Each sample is a rich, diacritized annotation object containing the following key fields:

- Audio: The raw waveform audio signal.

- Transcription: The original orthographic text of the Qur'anic verse.

- Phoneme: The target phoneme sequence for the utterance, generated using a specialized Arabic phonetizer (Kheir et al., 2025). This sequence serves as the primary learning target for phoneme-based recognition model.

3.1.2 Iqra'Eval dataset

The second stage of this work is trained and evaluated using the Iqra'Eval dataset [3], the provided dataset by the shared task, designed for Qur'anic Automatic Speech Recognition (ASR) and pronunciation evaluation. The dataset was utilized in the following predefined splits:

Training Split: Consists of 79 hours of audio. This partition contains a mixture of Modern Standard Arabic (MSA) speech and Qur'anic recitations, providing a curriculum-inspired foundation of general Arabic phonetics before specializing in the target domain.

Development Split: Comprises 3.4 hours of held-out Qur'anic recitations. This split was used exclusively for hyperparameter tuning, validation, and early stopping, ensuring a fair evaluation of the model's generalization capability. Each sample in the dataset follows key fields:

- Audio: The raw waveform audio signal.

- Sentence: The original orthographic text of the Qur'anic verse.

- Index: A unique identifier for the verse.

- Tashkeel_sentence: The fully diacritized text of the verse.

- Phoneme: The target phoneme sequence for the utterance. This sequence serves as the primary learning target for phoneme-based recognition models.

3.2 Testing Data

Final evaluation was the IqraEval Open Test datasetet[4]. This dataset is designed as a blind test set; it contains only audio data without ground truth transcriptions. The dataset consists of ≈ 2 h, with deliberate errors and human annotations, Predictions generated on this set are submitted to the Iqra'Eval organizers for evaluation scoring.

4 Results

4.1 Development Results

Stage-1 This curriculum setup helped stabilize training and improved the model's ability to generalize phoneme boundaries. After fine-tuning, the model achieved PER ≈ 0.54 on the held-out development data(EA-DI), establishing a strong baseline. **Stage-2** performed domain adaptation by further fine-tuning the stage-1 model on the Iqra_train dataset, which represents the official shared task domain. This stage achieved PER ≈ 0.21.

Model	Dataset	PER
Stage-1	EA-DI	0.54
Stage-2	Iqra_train	0.21

Table 1: Models Results on development set.

4.2 Open Test Results

Finally, beam search decoding was applied, yielding further gains at the sequence level. On the open test set, the best submission achieved an F1 score of 0.4236, with an accuracy of 0.8213 .

4.3 Qualitative Results

In this section a qualitative analysis is conducted based on examples from the development set. A comparative analysis of the outputs of both models againstst ground truth (Figure 1) indicates that deletions constitute the primary error type, with a limited number of perfect matches. Additionally, Figure 2 summarizes the character pairs that cause the most confusion.
Performance Statistics for 100 samples:

- Perfect Matches: 11 (11.0%)

- BS Improved: 11 (11.0%)

- BS Same Error: 68 (68.0%)

[3] https://huggingface.co/datasets/IqraEval/Iqra_train

[4] https://huggingface.co/datasets/IqraEval/open_testset

	F1-score(%)↑	Recall(%)↑	Precision(%)↑	CR(%)↑	Accu(%)↑	TA(%)↑	FR(%)↓	FA(%)↓	CD(%)↑
Baseline 1	**44.14**	**30.93**	77.07	83.61	**82.34**	**87.63**	**12.37**	22.93	**61.2**
Baseline 2	40.42	27.15	79.08	80.93	79.55	84.74	15.26	20.92	58.47
Stage-2	40.74	27.5	78.61	83.28	76.89	85.1	14.9	21.398	59.4
Stage-2 (BS)	42.36	28.79	**80.12**	**83.97**	82.13	85.75	14.25	**19.88**	60.3

Table 2: Experimental Results. ↓ Lower is better, ↑ Higher is better.

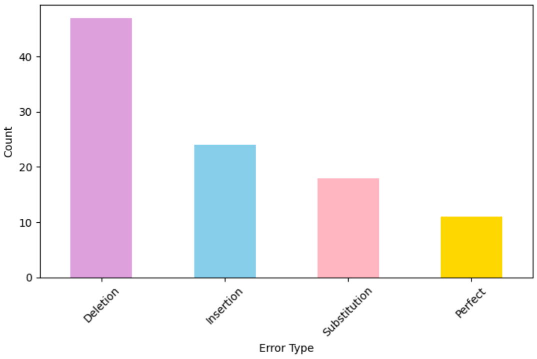


Figure 1: Error Type Distribution

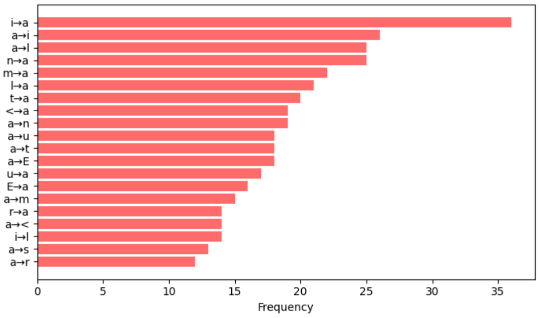


Figure 2: Top 20 Character Confusions (GT→Model)

- BS Worse: 10 (10.0%)

The error analysis reveals that the beam search decoding strategy provided minimal performance gains, often reproducing the same errors as the base model. In addition, a strong positive correlation was observed between the length of an utterance and the number of errors.

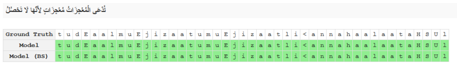
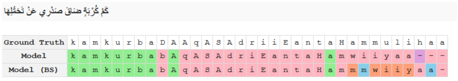
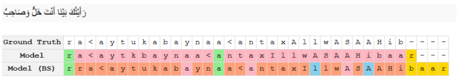
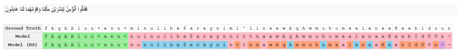
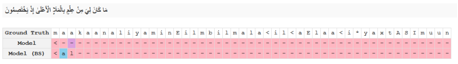
Sample	Results
1	 تُدْعَى الْمُعْجِزَاتُ مُعْجِزَاتٍ لِأَنَّهَا لَا تَحْصُلُ Ground Truth t u d E a a l m u E j i z a a t u m u E j i z a a t l i < a n n a h a a l a a t a H S U l Model t u d E a a l m u E j i z a a t u m u E j i z a a t l i < a n n a h a a l a a t a H S U l Model (BS) t u d E a a l m u E j i z a a t u m u E j i z a a t l i < a n n a h a a l a a t a H S U l
2	 كَمْ كُرْبَةٍ ضَاقَ صَدْرِي عَنْ تَحَمُّلِهَا Ground Truth k a m k u r b a D A A q A S A d r i i E a n t a H a m m u l i h a a Model k a m k u r b a b A q A S A d r i E a n t a H a m w i i y a a - - - Model (BS) k a m k u r b a b A q A S A d r i E a n t a H a m m m w i i y a a -
3	 رَأَيْتُكَ بَيْنَا أَنْتَ خَلٌّ وَصَاحِبٌ Ground Truth r a < a y t u k a b a y n a a < a n t a x A l l w A S A A H i b - - - - Model r a < a y t k b a y n a a < a n t a x I l l w A S A A H i b a a r - - - Model (BS) r r a < a y t u k a b a y n a a < a n t a x I l l w A S A A H i b a a r
4	 فَقَالُوا أَنُؤْمِنُ لِبَشَرَيْنِ مِثْلِنَا وَقَوْمُهُمَا لَنَا عَابِدُونَ Ground Truth f A q A A l u u < a n u < m i n u l i b a $ a r a y n i m i ^ l i n a a w A q A w m u h u m a a l a n a a E a a b i d u u n Model f A q A A l u u < a n u < n u i n u l i b a $ a r a y n n i t t n a a w A q A w m u h u m a a l a n a a E a a b I d U U n - Model (BS) f A q A A l u u < a n u < n u n u l i b a $ a r a y n i n t t n a a w A q A w m u h u m a a l a n a a E a a b I d U U n - -
5	 مَا كَانَ لِيَ مِنْ عِلْمٍ بِالْمَلَإِ الْأَعْلَىٰ إِذْ يَخْتَصِمُونَ Ground Truth m a a k a a n a l i y a m i n E i l m b i l m a l a < i l < a E l a a < i * y a x t A S I m u u n Model < - Model (BS) < a l -

Table 3: Comparison between Prediction results and Ground Truth, Green: Correct predictions, Light Red: Substitution errors, Blue: Characters corrected by Beam Search, Purple: Deletion errors, Gold: Insertion errors, Light Orange: Different errors in BS vs regular model

5 Discussion

Curriculum Learning. The staged approach validates curriculum-inspired fine-tuning (Bengio et al., 2009), as the general Qur'anic recitation training improved domain-specific adaptation.

Domain Adaptation. Without Stage-1, direct fine-tuning on IqraEval resulted in poor generalization. Adaptation through staged training aligns well with previous findings (Kunze and et al., 2017; Wang and et al., 2021).

Decoding. Beam search mitigated concatenation errors and improved phoneme sequences.

6 Conclusion

Two-stage fine-tuning pipeline was presented with domain adaptation and beam search decoding for Qur'anic ASR. Future work may include tajweed-aware decoding, phoneme-level language models, and adaptive curriculum schedules.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of ICML*.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Yassine El Kheir, Amit Meghanani, Hawau Olamide Toyin, Nada Almarwani, Omnia Ibrahim, Youssef Elshahawy, Mostafa Shahin, and Ahmed Ali. 2025. Iqra'eval: A shared task on qur'anic pronunciation assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Jonatas Grosman. 2021. Fine-tuned XLSR-53 large model for speech recognition in Arabic. `https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-arabic`.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Nawar Halabi. 2016. *Modern standard Arabic phonetics for speech synthesis*. Ph.D. thesis, University of Southampton.

Awni Hannun. 2017. Sequence modeling with ctc. *Distill*. Https://distill.pub/2017/ctc.

Yassine El Kheir, Ahmed Ali, and Shammur Absar Chowdhury. 2023. Automatic pronunciation assessment – a review. *Preprint*, arXiv:2310.13974.

Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Olamide Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, and 1 others. 2025. Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study. *arXiv preprint arXiv:2506.07722*.

Julius Kunze and et al. 2017. Transfer learning for speech recognition on a budget. In *Interspeech*.

Ambra Neri, Catia Cucchiarini, and Helmer Strik. 2008. The effectiveness of computer-based speech corrective feedback for improving segmental quality in l2 dutch. *ReCALL*, 20(2):225–243.

Emmanouil A Platanios, Otil Stretcu, Graham Neubig, Barnabás Póczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of NAACL*.

Changhan Wang and et al. 2021. Fine-tuning self-supervised speech models with limited data. In *ICASSP*.

IslamicEval 2025: The First Shared Task of Capturing LLMs Hallucination in Islamic Content

Hamdy Mubarak[1] , **Rana Malhas**[2], **Watheq Mansour**[3], **Abubakr Mohamed**[1],
Mahmoud Fawzi[4], **Majd Hawasly**[1], **Tamer Elsayed**[2], **Kareem Darwish**[1], **Walid Magdy**[4]
[1] QCRI, HBKU, Qatar
[2] Qatar University, Qatar
[3] The University of Queensland, Australia
[4] School of Informatics, The University of Edinburgh, UK
hmubarak@hbku.edu.qa, telsayed@qu.edu.qa, wmagdy@inf.ed.ac.uk

Abstract

Hallucination in Large Language Models (LLMs) remains a significant challenge and continues to draw substantial research attention. The problem becomes especially critical when hallucinations arise in sensitive domains, such as religious discourse. To address this gap, we introduce IslamicEval 2025—the first shared task specifically focused on evaluating and detecting hallucinations in Islamic content. The task consists of two subtasks: (1) Hallucination Detection and Correction of quoted verses (Ayahs) from the Holy Quran and quoted Hadiths; and (2) Qur'an and Hadith Question Answering, which assesses retrieval models and LLMs by requiring answers to be retrieved from grounded, authoritative sources. Thirteen teams participated in the final phase of the shared task, employing a range of pipelines and frameworks. Their diverse approaches underscore both the complexity of the task and the importance of effectively managing hallucinations in Islamic discourse.

1 Introduction

Large Language Models (LLMs) are becoming an integral part of natural language processing applications in Arabic. Recent advancements have produced several Arabic-focused and multilingual LLMs, such as Jais (Sengupta et al., 2023), Allam (Bari et al., 2024), and Fanar (Fanar Team et al., 2025), which have shown promising results across a variety of tasks, from open-domain question answering to content generation. However, alongside these advances, a critical challenge remains unresolved, namely hallucination, i.e. the generation of text that appears plausible but is factually incorrect or fabricated (Rawte et al., 2023).

This issue is particularly sensitive in domains where accuracy and authenticity are paramount, such as religion. In the Arabic-speaking world, religious topics are not only culturally central but also frequently searched, discussed, and queried online and in social media (Abokhodair et al., 2020; Fawzi et al., 2025), often driven by a deep sense of learning, curiosity, and at times, skepticism. This has made religious discourse, particularly question answering, among the most common applications of Arabic LLMs, both directly and indirectly.

Among religious sources, the Qur'an and Hadith literature stand out due to their sacred status and the high expectations of precision when they are quoted or referenced. The Qur'an, regarded as the ultimate and divine word of Allah, serves as the foundation of Islamic teachings. In tandem, Hadith encompasses the sayings, deeds, and implied approvals of the Prophet Muhammad (Peace Be Upon Him), serving in part as a practical illustration of Qur'anic teachings (Musallam, 2022). Given this sanctity, LLM hallucination in Islamic content poses serious risks: it can lead to misattributions, paraphrased verses falsely labeled as genuine, or entirely fabricated Hadiths (Fawzi et al., 2026), raising serious ethical, theological, and social concerns. Such hallucinations can unintentionally propagate misinformation or be exploited for disinformation, undermining trust in AI technologies and amplifying harm.

To address this gap, we have organized the IslamicEval 2025 shared task at ArabicNLP 2025,[1] which consists of two subtasks: (1) Hallucination Detection and Correction, and (2) Qur'an and Hadith Question Answering (QA). The first subtask focuses on detecting and correcting hallucinations in Qur'an and Hadith content within Arabic LLM-generated text. To our knowledge, it is the first task of its kind to target semantic and source-faithful evaluation of generated religious text. The second subtask is primarily intended to provide authentic QA benchmarks and standardized evalu-

[1] https://sites.google.com/view/islamiceval-2025/home

Proceedings of The Third Arabic Natural Language Processing Conference, pages 480–493
November 8-9, 2025 ©2025 Association for Computational Linguistics

ation testbeds for question answering models and systems on the Holy Qur'an and Hadith. Such benchmarks and testbeds are of paramount importance in the era of Generative AI, as they constitute a first line of defense against hallucination.

To this end, we aim to bring the Arabic NLP community together to develop robust systems for hallucination detection, localization, verification, and correction, as well as question answering on the Qur'an and Hadith:

Detection Determine whether a generated Arabic text contains a claimed Qur'anic verse (Ayah) or a Hadith. This involves building systems capable of semantic matching against canonical sources, accounting for variations in phrasing / paraphrasing.

Span Identification : Identify the exact span within the text corresponding to the claimed verse or Hadith. This requires models to accurately delimit religious content from surrounding context, often under noisy or stylistically varied conditions.

Verification Assess whether the detected quote is accurate—i.e., whether it exists in the authentic sources (e.g., Qur'an text or recognized Hadith collections) and is correctly cited. This step combines information retrieval with textual entailment techniques.

Correction If a quote is found to be inaccurate or hallucinated, reproduce the correct version, being the closest matching verse or Hadith if it exists, or indicate it is fabricated if no close match is found.

Passage Retrieval Given a free-text question in Modern Standard Arabic (MSA), a collection of Qur'anic passages covering the Holy Qur'an, and a collection of Hadiths from Sahih Al-Bukhari, the system must retrieve a ranked list of up to 20 answer-bearing passages—Qur'anic passages or Hadiths—that may contain one or more answers to the question, drawn from both collections.

This task raises unique NLP challenges:

- Fuzzy matching and paraphrase detection for verses and Hadiths expressed in non-standard forms;
- Robustness to stylistic variation and dialectal influence in generated text;
- Semantic grounding in authoritative religious corpora;
- Trust-sensitive evaluation, where false positives and false negatives have different and context-dependent implications.

We believe this shared task will catalyze research in faithful generation, hallucination detection, and knowledge-grounded NLP—not only for Arabic but as a reference for similar tasks in other languages and sensitive domains. It also supports the broader goal of responsible AI, promoting the development of LLMs that are not only fluent but also accurate, culturally aware, and ethically aligned.

2 IslamicEval Task 1: Hallucination Detection and Correction

Task 1 of the IslamicEval shared task addresses the detection and correction of hallucinations in LLM outputs that reference Qur'anic verses and Prophetic Hadiths. It is organized into three subtasks: identifying the intended references, validating their correctness against authoritative sources, and providing corrected versions when errors are found. The following subsections present the task setup, datasets, annotation guidelines, evaluation metrics, and results of participating systems.

2.1 Task Description

1. Subtask A - Identification of Intended verses (Qur'anic Ayahs) and Hadiths (Prophetic sayings) Given an LLM-generated response, participants will determine the spans of the "intended", since they might be inaccurate, verses and Hadiths in the text. Spans are represented by the character indexes, e.g. from character 0 to character 72 (inclusive). Evaluation is based on span precision and recall (macro-averaged F1 score). References to verse number and Hadith narrators and punctuations are ignored in this version.

2. Subtask B - Validation of content accuracy For each identified verse and Hadith, participants will categorize them as correct or incorrect based on established Islamic references. Evaluation is based on accuracy. Incorrect diacritics will be considered as mistakes.

3. Subtask C - Correction of Erroneous Content. Participants will provide corrected versions for any incorrectly quoted verse or Hadith, ensuring fidelity to the original sources. Evaluation is based on accuracy. Note that complete verse(s) from the Qur'an and complete Hadiths are expected. Writing and diacritics should be obtained from the shared Qur'an and Hadith sources.

2.2 Dataset

Starting from Qur'an QA 2023 dataset (n=251) that covers a broad range of topics including Fiqh, Tafsir, and Islamic teachings, a training (174), validation (25), and test (52) sets were created.

Six LLMs were prompted with the questions, with the prompt explicitly asking the models to cite Qur'anic and Hadith evidence in their responses (see Appendix B for the prompt). The question–output pairs, along with anonymized model IDs, were stored in XML format. The models used could be seen in Table 1. The LLM choice aimed to balance Arabic-focused models with state-of-the-art multilingual ones.

2.3 Annotation Setup and Guidelines

The generated answers were manually annotated by domain experts using the Label Studio platform (Tkachenko et al., 2020-2025). A separate annotation task was created for each question–response pair. Annotators were instructed to highlight every span containing an intended Qur'anic verse or Hadith and assign it one of four labels: Correct Qur'an, Incorrect Qur'an, Correct Hadith, or Incorrect Hadith. For each span marked as incorrect, annotators were required to either provide the corrected text or write "خطأ" (Wrong) if no valid correction existed. Figure 1 shows an example of an annotated output.

All annotators were experts in Islamic studies to ensure accuracy and reliability. Qur'anic references were standardized to the Uthmani script, while Hadith references were cross-checked against the six authoritative collections (الكتب الستة) including Sahih al-Bukhari and Sahih Muslim. The annotation guidelines emphasized precise span boundary selection and careful evaluation of correctness. The full annotation guidelines are available in Appendix C.

2.4 Evaluation Measures

Each subtask in Task 1 was evaluated using metrics suited to its specific objectives:

Subtask A (Identification) Performance was measured using the **macro-averaged F1** score, computed at the character level by classifying each character in the response string as belonging to a Qur'anic verse, a Hadith, or neither. Macro-averaged F1 is well-suited for this subtask because the data is highly imbalanced, with far fewer Ayah and Hadith spans than "neither", so accuracy alone would be misleading. Character-level F1 ensures that partial matches and boundary errors are fairly captured, while macro-averaging gives equal weight to each class rather than letting the dominant class overwhelm the results.

Subtask B (Validation) **Accuracy** was used as the evaluation metric, defined as the proportion of correctly assigned labels (Correct or Incorrect) over the total number of labels.

Subtask C (Correction) **Accuracy** was used, defined as the proportion of corrected outputs that exactly matched the corresponding ground truth over the total number of corrections. Strict accuracy was adopted for this subtask because even minor deviations - such as omitted words or altered diacritics - can substantially affect the meaning of a Qur'anic verse or Hadith. To avoid penalizing superficial formatting inconsistencies, both reference and system outputs were preprocessed prior to evaluation by removing default diacritics (e.g., sukun).

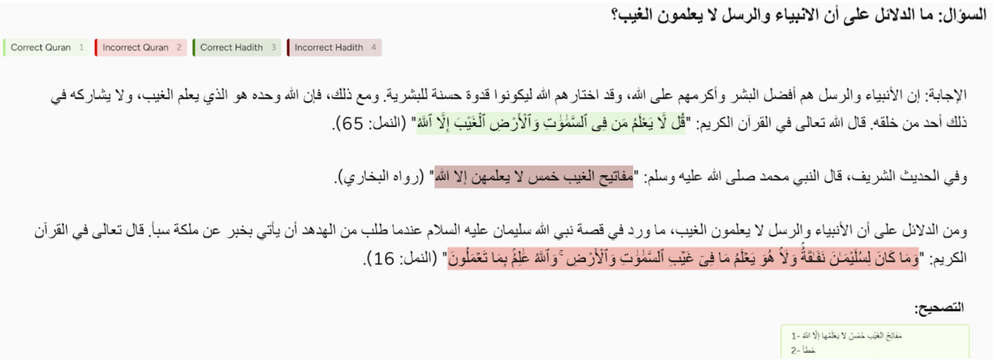
2.5 Task Setup

The dataset comprises 1,506 annotated answers (251 questions × 6 models). The development set corresponds to the original Qur'an QA 2023 dev set, consisting of 10% of the generations (n=150), yielding 50 annotated answers per subtask A, B and C. Similarly, the test set corresponds to the Qur'an QA 2023 test set, consisting of 20% of the questions (n=312), yielding 104 annotated answers per subtask. All annotations for development and test sets were manually reviewed and refined through multiple iterations (with the help of validation scripts) to ensure accuracy before release. A revised version of the training set (n=1,044) will be released in the future.

To facilitate participation, we hosted three competitions on CodaBench[2]. The development sets, along with the Qur'an and Hadith texts in JSON format (see Appendix D for a sample), were made publicly available. Participants were required to rely exclusively on the provided data.

The competition was launched on June 16, with test sets released on July 29, and final submissions closed on August 8. The shared task drew strong engagement, with 20 participants in Subtask 1A (87 submissions), 16 participants in Subtask 1B (41 submissions), and 15 participants in Subtask

[2] https://www.codabench.org/competitions/9820/, https://www.codabench.org/competitions/9822/, and https://www.codabench.org/competitions/9824/



السؤال: ما الدلائل على أن الانبياء والرسل لا يعلمون الغيب؟

الإجابة: إن الأنبياء والرسل هم أفضل البشر وأكرمهم على الله، وقد اختارهم الله ليكونوا قدوة حسنة للبشرية. ومع ذلك، فإن الله وحده هو الذي يعلم الغيب، ولا يشاركه في
ذلك أحد من خلقه. قال الله تعالى في القرآن الكريم: "قُل لَّا يَعْلَمُ مَن فِى ٱلسَّمَٰوَٰتِ وَٱلْأَرْضِ ٱلْغَيْبَ إِلَّا ٱللَّهُ" (النمل: 65).

وفي الحديث الشريف، قال النبي محمد صلى الله عليه وسلم: "مفاتيح الغيب خمس لا يعلمهن إلا الله" (رواه البخاري).

ومن الدلائل على أن الأنبياء والرسل لا يعلمون الغيب، ما ورد في قصة نبي الله سليمان عليه السلام عندما طلب من الهدهد أن يأتي بخبر عن ملكة سبأ. قال تعالى في القرآن
الكريم: "وَمَا كَانَ لِسُلَيْمَٰنَ نَفَقَةٌ وَلَا هُوَ يَعْلَمُ مَا فِى غَيْبِ ٱلسَّمَٰوَٰتِ وَٱلْأَرْضِ ۚ وَٱللَّهُ عَلِيمٌ بِمَا تَعْمَلُونَ" (النمل: 16).

التصحيح:

1- مَفَاتِحُ الغَيْبِ خَمْسٌ لا يَعْلَمُهَا إِلَّا اللهُ
2- خطأ

Figure 1: Example of an annotated LLM response. Question translation: "What is the evidence that the prophets and messengers do not know the unseen?". Spans highlighted in light green and dark green represent correct Qur'anic verses and Hadiths, respectively. Spans highlighted in light red and dark red represent incorrect Qur'anic verses and Hadiths. Corrections for each incorrect span are listed in the box at the bottom.

Model	#Answers	Avg Word Len	#Ayahs	Correct%	#Hadiths	Correct%
ALLaM-7B-Instruct-preview	251	297	1104	**84.06**	654	59.33
In-house fine-tuned Gemma-2-9b	251	153	548	65.33	372	38.17
In-house fine-tuned Gemma-2-9b + RAG*	246	742	1634	82.01	467	**63.17**
Jais-13B-Chat	251	46	151	41.72	83	26.51
Qwen3-8B	251	202	379	6.86	55	1.82
Llama-3.1-8B-Instruct	251	230	797	4.77	564	0.53

Table 1: Performance of models during dataset curation where LLM responses were annotated by experts. The model families include ALLam (Bari et al., 2024), Jais (Sengupta et al., 2023), Llama-3 (Grattafiori et al., 2024), Qwen3 (Qwen Team, 2025), in addition to fine-tuned versions of Gemma-2 (Gemma Team, 2024) developed in-house by the Fanar team (Fanar Team et al., 2025). Model marked with * failed to give answer to some questions. Best results in generating correct verses and Hadiths are written in bold.

1C (59 submissions). Since some teams submitted under multiple individual accounts, this amounted to five distinct teams overall, listed in Table 2.

Teams were allowed to submit an unlimited number of runs; however, only their most recent three submissions were considered for evaluation. Results were provided to participants on these final three runs, and they were requested to describe them in their system description papers.

2.5.1 Participating Teams

Burhan AI (Al Adel et al., 2025): For Subtask 1A, the authors fine-tuned a domain-adapted LLM (gpt-4.1-mini) for hallucination span detection, incorporating synthetic augmentation, diacritic variation, and morphological normalization to enhance robustness (F1 = 87.10%). In addition, they explored an agentic approach with specialized tools (OpenAI's code interpreter), achieving an **F1 of 90.06% (Best in subtask 1A)**. For Subtasks 1B (Accuracy = 88.60%) and 1C (Accuracy = 66.56%), they developed a multistage hierarchical correction pipeline that combined exact, normalized, fuzzy, and semantic matching with prompt-driven repair to ensure canonical alignment and diacritic fidelity.

HUMAIN (Omayrah et al., 2025): HUMAIN addressed Subtask 1 using a three-stage LLM-based pipeline grounded in the TANL framework (Paolini et al., 2021). For Subtask 1A, they modeled span detection as sequence-to-sequence annotation with bracket-based tags aligned via dynamic programming, with an alternative guided decoding setup through vLLM producing structured JSON. This achieved a strong 87.20% F1 on the test set. In Subtask 1B, validation combined retrieval-based similarity with strict substring matching, using higher thresholds for Qur'anic verses and exact substring logic for Hadith, yielding 86.14% accuracy. Finally, Subtask 1C correction employed multi-stage matching - exact, LCS alignment, and semantic reranking with bge-reranker-v2-m3 - reaching 68.18% accuracy, though rare Hadiths and implicit references remained challenging.

TCE (ElKoumy et al., 2025): The TCE team tackled Subtasks 1A and 1B of IslamicEval 2025 using few-shot prompting with state-of-the-art LLMs such as Qwen-235B (MoE) and GPT-4o. For span detection (1A), they used prompts enriched with trigger words and citation patterns, as well as chunking, and fuzzy matching to identify Qur'anic and Hadith content, achieving a macro-F1 of 86.11% on the test set. For 1B, they designed a retrieval-augmented pipeline: Qur'anic spans were retrieved with word-level fuzzy voting and Hadith with character n-gram TF-IDF, then verified by LLMs with strict word-for-word rules for Qur'an and lenient matching for Hadith. This hybrid system, enhanced with an efficient early-exit strategy, scored **89.82% accuracy (Best in Subtask 1B)** on the test set, with GPT-4o outperforming Gemma variants and showing improved performance when diacritics were preserved.

Isnad AI (Elden, 2025): The authors proposed a rule-based preprocessing and augmentation pipeline that systematically transforms raw religious texts into a large-scale, high-quality training corpus. The pipeline embeds processed Qur'anic verses and Hadiths into contextual templates. A set of common prefixes (eg. قال الله تعالى "God Almighty said") and suffixes (eg. رواه البخاري "Narrated by Al-Bukhari") was applied, and each unique instance was expanded into multiple training examples by randomly combining it with different prefixes, suffixes, and neutral connecting sentences. The authors reported that synthetic data generation using AraGPT was less effective.

2.5.2 Task 1 Results

Table 2 shows the results for Task 1 across the three subtasks. Participating systems employed a wide range of approaches to detect the intended Qur'anic verses and Hadiths, including LLMs such as GPT-4 and Qwen, as well as fuzzy matching with search engines and rule-based techniques. Our evaluation shows a significant performance gap: the rule-based approach (e.g. Isnad AI) lag considerably behind LLM-based systems, highlighting the inherent difficulty of this task. Lists of rules and patterns are insufficient to capture the diverse styles and degrees of distortion found in LLM generations.

We also observe that detecting the textual boundaries of verses and Hadiths is substantially easier than correcting them, underscoring the fact that hallucinations in LLM outputs are often non-trivial to repair. Recovery from hallucinated references remains highly challenging, suggesting that hallucination prevention should occur during generation, e.g. via RAG to constrain outputs to authentic sources, instead of post-hoc correction.

Finally, we find that models perform consistently better on Qur'anic verses than on Hadiths (either by the participating teams or the LLMs in Table 1). This can be attributed to the relative size and structure of the corpora: the Qur'an is comparatively compact and standardized, whereas Hadith collections (e.g., the six authoritative books) are far larger and more variable, making hallucination detection and correction more complex.

3 IslamicEval Task 2: Qur'an and Hadith Question Answering

In this section, we define Task 2, its dataset, annotation and evaluation setup, and the measures used to rank systems. Results are presented and discussed before concluding with an overview of the approaches adopted by the systems of participating teams (with accepted description papers).

3.1 Task Description

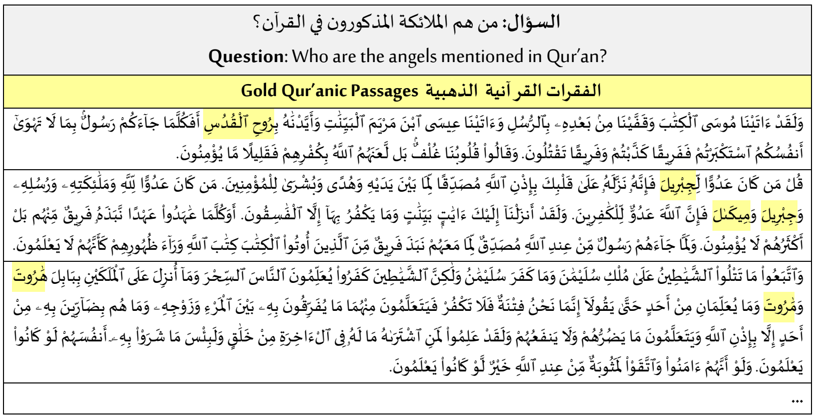
The Qur'an and Hadith QA subtask is a continuation of the Qur'an QA 2022[3] (Malhas et al., 2022) and Qur'an QA 2023[4] (Malhas et al., 2023) Shared Tasks. This year's subtask introduces Hadith as an additional Islamic resource for answering questions, marking the first such inclusion in the task's history. We define the task as follows: Given a free-text question in Modern Standard Arabic (MSA), a collection of Qur'anic passages covering the Holy Qur'an, and a collection of Hadiths from Sahih Al-Bukhari, systems are required to retrieve a ranked list of up to 20 answer-bearing Qur'anic passages or Hadiths (i.e., that potentially contain the answer(s) to the given question) drawn from these two collections. Questions may be factoid or non-factoid. Example questions with answer-bearing Qur'anic passages and Hadith *matns* are exhibited in Figures 2 and 3, respectively. The *matn* refers to the core text of the Hadith itself, while the *isnad* outlines the chain of narrators who convey and authenticate the *matn* (Azmi et al., 2019).

To better reflect real-world conditions and make the task more challenging, we included questions that lack answers in the Qur'an and/or Sahih Al-Bukhari. We label a question *zero-answer* only when neither source contains an answer. For such questions, the ideal system returns no result; otherwise, it should output a ranked list of up to 20 answer-bearing Qur'anic passages or Hadith *matns*.

[3] https://sites.google.com/view/quran-qa-2022
[4] https://sites.google.com/view/quran-qa-2023

	Subtask 1A				Subtask 1B				Subtask 1C			
Team Name	**F1%**	**F1-Q**	**F1-H**	**Rank**	**Acc%**	**Acc-Q**	**Acc-H**	**Rank**	**Acc%**	**Acc-Q**	**Acc-H**	**Rank**
Burhan AI	**90.06**	89.47	86.99	**1**	88.60	89.45	86.63	2	66.56	65.70	67.65	2
HUMAIN	87.20	86.61	85.11	2	86.14	90.20	76.74	3	**68.18**	62.21	75.74	**1**
TCE	86.11	86.60	80.51	3	**89.82**	91.21	86.63	**1**	-	-	-	-
Isnad AI	66.97	72.39	48.94	4	-	-	-	-	-	-	-	-
mucAI*	44.88	46.24	29.80	5	-	-	-	-	-	-	-	-

Table 2: Task 1 results across subtasks. Teams are ranked per subtask. The majority baseline in Subtask 1A is **36.17%** Macro-Avg. F1 (assuming no Ayah or Hadith), in Subtask 1B is **70.00%** (assuming all Ayahs and Hadiths are correct), and in Subtask 1C is **67.52%** (assuming all errors are not correctable). For Subtask 1A we report the overall Macro-averaged F1 (F1) and for Qur'an (F1-Q) and Hadith (F1-H) individually. Similarly, for Subtasks 1B and 1C we report the accuracies Acc, Acc-Q, and Acc-H. Teams marked with * did not submit a system paper.



السؤال: من هم الملائكة المذكورون في القرآن؟

Question: Who are the angels mentioned in Qur'an?

Gold Qur'anic Passages الفقرات القرآنية الذهبية

وَلَقَدْ ءَاتَيْنَا مُوسَى ٱلْكِتَٰبَ وَقَفَّيْنَا مِنۢ بَعْدِهِۦ بِٱلرُّسُلِ وَءَاتَيْنَا عِيسَى ٱبْنَ مَرْيَمَ ٱلْبَيِّنَٰتِ وَأَيَّدْنَٰهُ بِرُوحِ ٱلْقُدُسِ أَفَكُلَّمَا جَآءَكُمْ رَسُولٌۢ بِمَا لَا تَهْوَىٰٓ
أَنفُسُكُمُ ٱسْتَكْبَرْتُمْ فَفَرِيقًا كَذَّبْتُمْ وَفَرِيقًا تَقْتُلُونَ. وَقَالُوا۟ قُلُوبُنَا غُلْفٌۢ بَل لَّعَنَهُمُ ٱللَّهُ بِكُفْرِهِمْ فَقَلِيلًا مَّا يُؤْمِنُونَ.

قُلْ مَن كَانَ عَدُوًّا لِّجِبْرِيلَ فَإِنَّهُۥ نَزَّلَهُۥ عَلَىٰ قَلْبِكَ بِإِذْنِ ٱللَّهِ مُصَدِّقًا لِّمَا بَيْنَ يَدَيْهِ وَهُدًى وَبُشْرَىٰ لِلْمُؤْمِنِينَ. مَن كَانَ عَدُوًّا لِّلَّهِ وَمَلَٰٓئِكَتِهِۦ وَرُسُلِهِۦ
وَجِبْرِيلَ وَمِيكَىٰلَ فَإِنَّ ٱللَّهَ عَدُوٌّ لِّلْكَٰفِرِينَ. وَلَقَدْ أَنزَلْنَآ إِلَيْكَ ءَايَٰتٍۭ بَيِّنَٰتٍ وَمَا يَكْفُرُ بِهَآ إِلَّا ٱلْفَٰسِقُونَ. أَوَكُلَّمَا عَٰهَدُوا۟ عَهْدًا نَّبَذَهُۥ فَرِيقٌ مِّنْهُم بَلْ
أَكْثَرُهُمْ لَا يُؤْمِنُونَ. وَلَمَّا جَآءَهُمْ رَسُولٌ مِّنْ عِندِ ٱللَّهِ مُصَدِّقٌ لِّمَا مَعَهُمْ نَبَذَ فَرِيقٌ مِّنَ ٱلَّذِينَ أُوتُوا۟ ٱلْكِتَٰبَ كِتَٰبَ ٱللَّهِ وَرَآءَ ظُهُورِهِمْ كَأَنَّهُمْ لَا يَعْلَمُونَ.

وَٱتَّبَعُوا۟ مَا تَتْلُوا۟ ٱلشَّيَٰطِينُ عَلَىٰ مُلْكِ سُلَيْمَٰنَ وَمَا كَفَرَ سُلَيْمَٰنُ وَلَٰكِنَّ ٱلشَّيَٰطِينَ كَفَرُوا۟ يُعَلِّمُونَ ٱلنَّاسَ ٱلسِّحْرَ وَمَآ أُنزِلَ عَلَى ٱلْمَلَكَيْنِ بِبَابِلَ هَٰرُوتَ
وَمَٰرُوتَ وَمَا يُعَلِّمَانِ مِنْ أَحَدٍ حَتَّىٰ يَقُولَآ إِنَّمَا نَحْنُ فِتْنَةٌ فَلَا تَكْفُرْ فَيَتَعَلَّمُونَ مِنْهُمَا مَا يُفَرِّقُونَ بِهِۦ بَيْنَ ٱلْمَرْءِ وَزَوْجِهِۦ وَمَا هُم بِضَآرِّينَ بِهِۦ مِنْ
أَحَدٍ إِلَّا بِإِذْنِ ٱللَّهِ وَيَتَعَلَّمُونَ مَا يَضُرُّهُمْ وَلَا يَنفَعُهُمْ وَلَقَدْ عَلِمُوا۟ لَمَنِ ٱشْتَرَىٰهُ مَا لَهُۥ فِى ٱلْءَاخِرَةِ مِنْ خَلَٰقٍ وَلَبِئْسَ مَا شَرَوْا۟ بِهِۦٓ أَنفُسَهُمْ لَوْ كَانُوا۟
يَعْلَمُونَ. وَلَوْ أَنَّهُمْ ءَامَنُوا۟ وَٱتَّقَوْا۟ لَمَثُوبَةٌ مِّنْ عِندِ ٱللَّهِ خَيْرٌ لَّوْ كَانُوا۟ يَعْلَمُونَ.

...

Figure 2: An example question with some of its gold (answer-bearing) Qur'anic passages. Answers are highlighted.

3.2 Dataset

In this section, we introduce the test collections used for the Qur'an-Hadith QA subtask (or QH-QA for short). In information retrieval, a *test collection* consists of a document collection[5] (here, the Holy Qur'an and Sahih al-Bukhari), a set of queries (questions), and their relevance judgments (Lin and Katz, 2006) (i.e., the gold answers or, in our case, the passages that contain them).

The document collections used for this subtask comprise the Qur'anic Passage collection (QPC) (Swar, 2007; Malhas, 2023), and Sahih Al-Bukhari collection. QPC was developed by topically segmenting the 114 Qur'anic chapters using the Thematic Holy Qur'an (Swar, 2007)[6], a printed edition that clusters the chapter verses into topics. This segmentation resulted in a total of 1,266 passages. For the Sahih Al-Bukhari collection, we used the Tajreed Sarih version (Al-Zubaidi, 2009) that comprises 2,254 Hadiths, from which redundant Hadiths, Arabic commentary, and chain of narrators (except the last) have been excluded. However, Al-Zubaidi may repeat a Hadith if there was a beneficial addition in a later occurrence. Moreover, only authenticated Hadiths with a continuous chain of narrators are included in this collection. The digital version of this book[7] is available on `shamela.ws`, a project for collecting classical Arabic books. We contacted an Islamic scholar who provided us with an offline version of the book, which we parsed later to generate the final JSON lines *(.jsonl)* format[8].

For the questions, we used the 250 questions of *AyaTEC v1.2* dataset (Malhas and Elsayed, 2020; Malhas et al., 2023), split into training (84%) and development (16%) sets. The relevance judgments for these questions are provided over the QPC **only**.

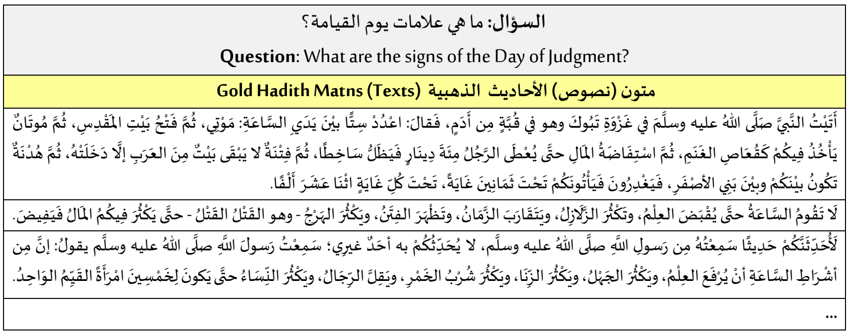
For the test dataset, we developed a new set of 71 questions, 23 of which are paraphrased versions of natural user prompts drawn from usage logs of the Fanar Arabic LLM (Fanar Team et al., 2025). Only 51 questions were used to evaluate the systems of participating teams. The relevance judgments for all 71 questions over the Qur'anic Passage col-

[5] The term "document collection" or "collection" refers to a corpus or dataset (Lin et al., 2021); we use these terms interchangeably.

[6] `https://archive.org/details/Quran27/page/n13/mode/2up`

[7] `shamela.ws/book/96283/`

[8] `https://gitlab.com/bigirqu/quran-hadith-qa-2025`



السؤال: ما هي علامات يوم القيامة؟ **Question**: What are the signs of the Day of Judgment?
Gold Hadith Matns (Texts) متون (نصوص) الأحاديث الذهبية
أَتَيْتُ النَّبِيَّ صَلَّى اللهُ عليه وسلَّمَ في غَزْوَةِ تَبُوكَ وهو في قُبَّةٍ مِن أَدَمٍ، فَقالَ: اعْدُدْ سِتًّا بيْنَ يَدَيِ السَّاعَةِ: مَوْتِي، ثُمَّ فَتْحُ بَيْتِ المَقْدِسِ، ثُمَّ مُوتَانٌ يَأْخُذُ فِيكُمْ كَقُعَاصِ الغَنَمِ، ثُمَّ اسْتِفَاضَةُ المَالِ حتَّى يُعْطَى الرَّجُلُ مِئَةَ دِينَارٍ فَيَظَلُّ سَاخِطًا، ثُمَّ فِتْنَةٌ لا يَبْقَى بَيْتٌ مِنَ العَرَبِ إلَّا دَخَلَتْهُ، ثُمَّ هُدْنَةٌ تَكُونُ بيْنَكُمْ وبيْنَ بَنِي الأصْفَرِ، فَيَغْدِرُونَ فَيَأْتُونَكُمْ تَحْتَ ثَمَانِينَ غَايَةً، تَحْتَ كُلِّ غَايَةٍ اثْنَا عَشَرَ أَلْفًا.
لَا تَقُومُ السَّاعَةُ حتَّى يُقْبَضَ العِلْمُ، وتَكْثُرَ الزَّلَازِلُ، ويَتَقَارَبَ الزَّمَانُ، وتَظْهَرَ الفِتَنُ، ويَكْثُرَ الهَرْجُ - وهو القَتْلُ القَتْلُ - حتَّى يَكْثُرَ فِيكُمُ المَالُ فَيَفِيضَ.
لَأُحَدِّثَنَّكُمْ حَدِيثًا سَمِعْتُهُ مِن رَسولِ اللَّهِ صلَّى اللهُ عليه وسلَّم، لا يُحَدِّثُكُمْ به أحَدٌ غيرِي: سَمِعْتُ رَسولَ اللَّهِ صلَّى اللهُ عليه وسلَّم يقولُ: إنَّ مِن أَشْرَاطِ السَّاعَةِ أنْ يُرْفَعَ العِلْمُ، ويَكْثُرَ الجَهْلُ، ويَكْثُرَ الزِّنَا، ويَكْثُرَ شُرْبُ الخَمْرِ، ويَقِلَّ الرِّجَالُ، ويَكْثُرَ النِّسَاءُ حتَّى يَكونَ لِخَمْسِينَ امْرَأَةً القَيِّمُ الوَاحِدُ.
...

Figure 3: An example question with some of its gold (answer-bearing) Hadith *matns* from Sahih Al-Bukhari.

lection and the Sahih Al-Bukhari collection were conducted by Qur'an and Hadith specialists, as described in the next section.

We note that the relevance judgments for the test dataset will not be released. Nevertheless, future run submissions for evaluation on this dataset may be obtained by contacting one of the organizers. All datasets and test collections are publicly available in the official Qur'an-Hadith QA repository.[9]

3.3 Annotation Setup and Guidelines

Two annotation guidelines and rubrics, with illustrative examples, were meticulously developed for the Qur'an and Hadith specialists, labeling potential answer-bearing Quranic passages and Hadith *matns*. Each candidate passage or *matn* was annotated as either having a *direct answer*, an *indirect answer*, *relevant but no answer*, or *irrelevant* to a given question. The Arabic definitions for these labels are in Figures 7 and 8 (Appendix E).

Moreover, Arabic web-based GUIs were developed in line with these guidelines and rubrics to streamline annotation and gather specialist-suggested passages and *matns* potentially containing *direct* or *indirect answers* to the given question.

Retrieval and pooling: We constructed a pooled candidate set per question by taking the deduplicated union of top-k results from multiple retrieval models. The pooled candidates were re-ranked using GPT-4.1 and GPT-4.1-mini. We applied a cutoff at the top-20 items after re-ranking to define the Round 1 candidate set presented to annotators.

Annotation rounds and coverage. In Round 1, specialists annotated the re-ranked top-20 candidates per question (across both collections). In Round 2, they annotated additional candidates that they had proposed during Round 1. Round 3 took place after the test-set submission phase closed, during which specialists annotated a pooled candidate set per question, formed as the deduplicated union of the top-k responses from the best submitted run of each team, after excluding candidates with a frequency less than 2.

Each candidate passage/matn in Round 1 was independently labeled by three Qur'an specialists (for Qur'anic passages) or three Hadith specialists (for Hadith *matns*). Additional candidates in Rounds 2 and 3 were likewise independently labeled by three domain specialists.

Aggregation and agreement: We applied majority voting across the three domain specialists; ties were resolved by a fourth. Despite our careful design and piloting of the annotation rubrics, inter-annotator agreement was fair: Fleiss' kappa was 0.283 among Qur'an specialists and 0.235 for Hadith.

Label normalization: Consistent with the training and development sets, final test-set relevance judgments over both collections were binarized: only passages/matns containing a *direct answer* received a positive label (1); all others received 0.

3.4 Evaluation Setup

We chose Codabench[10] as a platform for hosting our subtask, similar to Task 1. We used trec_eval tool[11] to compute the evaluation metrics. We made our training and development sets available during the development phase and allowed each team to run 100 submissions on the development set and receive scores from the system. Our evaluation script was also made available for local evaluation. During the testing phase, we allowed teams to submit 13 submissions; however, we stated that only the last 3 submitted runs would be considered for evaluation.

[9] https://gitlab.com/bigirqu/quran-hadith-qa-2025

[10] codabench.org/competitions/9939/

[11] github.com/usnistgov/trec_eval

Team	MAP@10	MAP_Q@5	MAP_H@5
Burhan	**0.3351**	**0.3389**	**0.3876**
BurhanAI	0.2807	0.3257	0.2386
ThinkDrill	0.2296	0.2623	0.215
NUR	0.1809	0.2334	0.1923
BayaNet*	0.1504	0.2064	0.224
MSA*	0.1185	0.1674	0.0685
Maged*	0.0332	0.0887	0.0457
CISRG*	0.0116	0.0294	0.0128

Table 3: Results of Task 2 showing the best run per team ranked by MAP@10. Teams with * did not submit a system paper.

3.4.1 Evaluation Measures

For the classical ranked retrieval formulation of the task, MAP (Mean Average Precision) serves as the primary official evaluation metric. The no-answer cases are handled simply by giving full credit to "no answers" system output and zero otherwise. We report three measures: **MAP@10** computed over the top 10 ranked answers, **MAP_Q@5** computed over the top 5 ranked Qur'anic passages (after discarding all ranked Hadiths), and **MAP_H@5** computed over the top 5 ranked Hadiths (after discarding all ranked Qur'anic passages).

3.4.2 Participating Teams and Results

While 30 teams registered in Task 2, eight teams submitted runs during the test phase. The evaluation of the best run per team is shown in Table 3. For the full evaluation results, see Table 4 in Appendix. Only four out of eight participating teams submitted papers describing their work, namely Burhan (Basheer et al., 2025), ThinkDrill (Elrefai et al., 2025), Nur (Amin et al., 2025), and BurhanAI (Al Adel et al., 2025). It is evident that the task of this year is quite challenging since the top MAP@10 score is 0.3351 achieved by Burhan.

3.5 Methods and Analysis

The main observation in all participants is the reliance on LLMs in their systems. We categorize the discussion of adopted methods by techniques.

Augmentation The top team (Burhan) utilized LLMs to extract facts and relationships from Qur'an and Hadith passages and then augmented the extracted text with the corresponding passages. ThinkDrill team extended hadith question-answer pairs from HAQA dataset, and employed GPT-4 to extract relevant keywords from questions, and then apply fuzzy string matching to determine the relevance score. NUR team augmented the provided dataset with the Arabic portion of the TyDi dataset, the Jalalayn Tafseer of the Qur'an, and the QuQA and HaQA datasets. They also embedded negative samples to increase their models' sensitivity to zero-answer questions. BurhanAI team employed iterative semantic search, expanding the query with the initial results.

Reranking Burhan and ThinkDrill adopted an LLM as a reranker, leading to remarkable improvements as reported by Burhan team. NUR team used a fine-tuned cross-encoder or Gemini for reranking and identification of zero-answer questions.

Embedding Toward building sematic-based retrieval pipelines, multiple teams focused on the choice of the encoder embedding model. Burhan team experimented with multiple embedding models to identify the best model in *zero-shot* setup. However, ThinkDrill *fine-tuned* a multilingual embedding model using triplet loss on augmented data of Qur'an and Hadith. NUR team has compared a large set of publicly available Arabic sentence embedding models on the development set (Qur'an-only) to select the backbone encoder for their retrieval and reranking pipeline. On the other hand, BurhanAI team employed OpenAI's `file_search` directly as the backbone for semantic search.

Paraphrasing Burhan team was the only team that worked on improving the query representation. In particular, they utilize LLMs to paraphrase the questions or append synonyms to them. The paraphrasing component revealed clear benefits.

Zero-answer Questions Handling the zero-answer questions differed across teams. Burhan team employed an LLM to judge whether a passage provides an answer to a given question on a binary basis. ThinkDrill adopted a thresholding mechanism to detect such questions, i.e., if the relevance score is above s certain threshold, the question then has an answer. Similarly, NUR team adopted the thresholding-based approach with fine-tuned cross-encoders, in addition to directly prompting Gemini LLM to identify such questions.

4 Conclusion

We introduced IslamicEval, the first shared task dedicated to addressing hallucination in Islamic contexts. The challenges posed by this task aim to significantly advance the reliability of LLMs in generating accurate Islamic content. Moreover, it supports broader efforts to uphold the integrity of religious information in the digital age.

5 Limitations

Labeling religious data is an exhaustive sensitive task. As a result, the number of records in our datasets is not big. We plan in the future to extend our datasets by labeling more samples.

Our study only considers Qur'an and Hadith in the Arabic language; however, there are hundreds of millions of people worldwide who communicate Hadith in other languages like Turkish, Farsi, Malay, and Urdu (Fawzi et al., 2026). Since these languages have their own customized LLMs, it is very likely that they will produce different variants of religious hallucinations. In addition, each LLM output in Subtask 1 was annotated by a single annotator, which may introduce annotation errors. We evaluated answers from six LLMs (Arabic-centric and multilingual), each with distinct styles of responding to Islamic questions, which may not generalize to other models. The test set is relatively small (312 question–answer pairs), and model performance could vary on larger or thematically different test sets. Furthermore, annotation was limited to assessing the correctness of Qur'anic verses and Hadiths, without considering whether the overall answer was accurate or relevant to the input question. A more comprehensive evaluation of LLMs in this domain should therefore extend beyond text correction to include additional dimensions of answer quality.

Since this is the first edition of the Qur'an–Hadith QA task to incorporate Hadith as an additional Islamic resource for answering questions, we limited the Hadith collection to Sahih al-Bukhari. We plan to include other Hadith collections in future versions of the task.

Unlike the AyaTEC and QRCD datasets used in prior versions of Subtask 2, the annotation phase for the current test set may not have exhaustively identified all answer-bearing candidates. Consequently, evaluation is subject to the usual risk that some relevant results may not be rewarded.

6 Ethical Considerations

Subtask 1 involves questions and answers generated by LLMs, which were manually annotated to correct errors in cited Qur'anic verses and Hadiths. Given the religious sensitivity of the content, we took care to ensure accuracy and respect: annotations were carried out by three qualified linguists from Egypt with expertise in Arabic language and Islamic studies, and all annotators were compensated fairly for their work. The dataset is released strictly for research purposes, with the intention of improving the reliability and safety of LLMs in handling religious material. We explicitly caution against any misuse of this resource in contexts that could distort, misrepresent, or disrespect Islamic teachings.

7 Acknowledgments

We gratefully acknowledge the Qur'an and Hadith specialists who helped fine-tune test-question language, evaluated and annotated answers from the Qur'an and Sahih al-Bukhari, and advised on Hadith resources for Subtask 2—especially the professors of the Department of Qur'an and Sunnah, College of Sharia and Islamic Studies, Qatar University: Prof. Ahmad Shukri, Dr. Abdulhamid Alsis, and Dr. Abada Tahhan. The work of Watheq Mansour would not have been possible without the support of resources from Dr. Joel Mackenzie.

References

Samir Abdaljalil, Hasan Kurban, and Erchin Serpedin. 2025. Halluverse25: Fine-grained multilingual benchmark dataset for llm hallucinations. *arXiv preprint arXiv:2503.07833*.

Norah Abokhodair, AbdelRahim Elmadany, and Walid Magdy. 2020. Holy tweets: Exploring the sharing of the quran on twitter. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–32.

Basem H. Ahmed, Motaz K. Saad, and Eshrag A. Refaee. 2022. QQATeam at Quran QA 2022: Fine-Tuning Arabic QA Models for Quran QA Task. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*.

Arij Al Adel, Abu Bakr Soliman, Mohamed Sakher Sawan, Rahaf Al-Najjar, and Sameh Amin. 2025. Combating hallucinations in llms for islamic content: Evaluation, correction, and retrieval-based solution. In *Proceedings of the Third Arabic Natural Language Processing Conference*.

Hend Al-Khalifa, Tamer Elsayed, Hamdy Mubarak, Abdulmohsen Al-Thubaity, Walid Magdy, and Kareem Darwish. 2022. Proceedinsg of the 5th workshop on osact with shared tasks on qur'an qa and fine-grained hate speech detection. In *Proceedinsg 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*.

Z. A. B. Al-Zubaidi. 2009. *Al-Tajreed Al-Sareeh of Collective Sahih Hadith*. Resalah Publishers. Author died 893 AH/1488 CE.

Hayfa A Aleid and Aqil M Azmi. 2025. Hajj-fqa: A benchmark arabic dataset for developing question-answering systems on hajj fatwas: H. aleid and a. azmi. *Journal of King Saud University Computer and Information Sciences*, 37(6):135.

Serag Amin, Ranwa Aly, Yara Allam, Yomna Eid, and Ensaf Hussein Mohamed. 2025. Nur at islamiceval 2025 shared task: Retrieval-augmented llms for qur'an and hadith qa. In *Proceedings of the Third Arabic Natural Language Processing Conference*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Araelectra: Pre-training text discriminators for arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195. Association for Computational Linguistics.

Aqil M Azmi, Abdulaziz O Al-Qabbany, and Amir Hussain. 2019. Computational and natural language processing based studies of hadith literature: a survey. *Artificial Intelligence Review*, 52:1369–1414.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan AlRashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, et al. 2024. ALLam: Large language models for Arabic and English. *arXiv preprint arXiv:2407.15390*.

Mohammad Basheer, Watheq Mansour, Abdulhamid Touma, and Ahmad Qadeib Alban. 2025. Burhan at islamiceval: Fact-augmented and llm-driven retrieval for islamic qa. In *Proceedings of the Third Arabic Natural Language Processing Conference*.

Amina El Ganadi, Sania Aftar, Luca Gagliardelli, Federico Ruozzi, et al. 2025. Generative ai for islamic texts: The eman framework for mitigating gpt hallucinations. In *roceedings of the 17th International Conference on Agents and Artificial Intelligence-ICAART*, volume 3, pages 1221–1228.

Fatimah Mohamed Emad Elden. 2025. Isnad ai at islamiceval 2025: A rule-based system for identifying religious texts in llm outputs. In *Proceedings of the Third Arabic Natural Language Processing Conference*.

Mohammed ElKomy and Amany M. Sarhan. 2022. TCE at Qur'an QA 2022: Arabic Language Question Answering Over Holy Qur'an Using a Post-Processed Ensemble of BERT-based Models. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*.

Mohammed Alaa Elkomy and Amany Sarhan. 2023. TCE at Qur'an QA 2023 Shared Task: Low Resource Enhanced Transformer-based Ensemble Approach for Qur'anic QA. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Mohammed ElKoumy, Mohamed Ibrahim Alqablawi, Ahmad Tamer, and Khalid Allam. 2025. Tce at islamiceval 2025: Retrieval-augmented llms for quranic and hadith content identification and verification. In *Proceedings of the Third Arabic Natural Language Processing Conference*.

Eman Elrefai, Toka Khaled, and Ahmed Soliman. 2025. Thinkdrill at islamiceval 2025: Llm hybrid approach for quran and hadith question answering. In *Proceedings of the Third Arabic Natural Language Processing Conference*.

Fanar Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, et al. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Mahmoud Fawzi, Walid Magdy, and Björn Ross. 2025. 'the prophet said so!': On exploring hadith presence on arabic social media. *Proceedings of the ACM on Human-Computer Interaction*, 9(2):1–23.

Mahmoud Fawzi, Björn Ross, and Walid Magdy. 2026. Fabricating holiness: Characterizing religious misinformation circulators on arabic social media. *Proceedings of the International AAAI Conference on Web and Social Media*, 20.

Gemma Team. 2024. Gemma.

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, and Arun Rao et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

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.

Jimmy Lin and Boris Katz. 2006. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology*, 57(7):851–861.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.

Rana Malhas and Tamer Elsayed. 2020. AyaTEC: Building a Reusable Verse-based Test Collection for Arabic Question Answering on the Holy Qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 19(6):1–21.

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 (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*, pages 79–87.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Rana R Malhas. 2023. *ARABIC QUESTION ANSWERING ON THE HOLY QUR'AN*. Ph.D. thesis, Qatar University.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*.

Hamdy Mubarak, Hend Al-Khalifa, and Khaloud Suliman Alkhalefah. 2024. Halwasa: Quantify and analyze hallucinations in large language models: Arabic as a case study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8008–8015.

Al-Tahir A.R. Musallam. 2022. Prophetic interpretation of the quran: Between quantity and quality. *Rehan Journal for Scientific Publishing*, 26(1):51–76.

Arwa Omayrah, Sakhar Alkhereyf, Ahmed Abdelali, Abdulmohsen Al-Thubaity, Jeril Kuriakose, and Ibrahim AbdulMajeed. 2025. Humain at islamiceval 2025 shared task 1: A three-stage llm-based pipeline for detecting and correcting hallucinations in quran and hadith. In *Proceedings of the Third Arabic Natural Language Processing Conference*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *arXiv preprint arXiv:2101.05779*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, et al. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Marwan N. Swar. 2007. *Mushaf Al-Tafseel Al-Mawdoo'ee*. Dar Al-Fajr Al-Islami, Damascus.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2025. Label Studio: Data labeling software. Open source software available from https://github.com/HumanSignal/label-studio.

Raúl Vázquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sánchez-Vega, Alessandro Raganato, Jindřich Libovickỳ, et al. 2025. Semeval-2025 task 3: Mu-shroom, the multilingual shared task on hallucinations and related observable overgeneration mistakes. *arXiv preprint arXiv:2504.11975*.

Abdulrezzak Zekiye and Fadi Amroush. 2023. Al-jawaab at Qur'an QA 2023 shared task: Exploring Embeddings and GPT Models for Passage Retrieval and Reading Comprehension. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

A Related Work

A.1 Hallucination Detection

Hallucination detection methods can be grouped into uncertainty-based predictors (Manakul et al., 2023), entailment or consistency checks against retrieved evidence (Ji et al., 2023), and span-level labeling frameworks (Mishra et al., 2024). Recent work emphasizes span-level detectors for interpretability, with SemEval-2025 introducing a shared task that explicitly included Arabic (Vázquez et al., 2025).

For Arabic hallucination detection, resources remain limited. The OSACT-6 Hallucination Shared Task "Halwasa" (Mubarak et al., 2024) released the first Arabic data set (10K sentences generated by GPT and manually annotated for factuality), with baselines that highlight challenges due to morphological richness. HalluVerse25 (Abdaljalil et al., 2025) is a multilingual benchmark that categorizes fine-grained hallucinations in English, Arabic, and Turkish. The authors used GPT-4 to inject hallucinations into factual biographical sentences extracted from Wikipedia.

In religious domains, hallucination risks are amplified by doctrinal sensitivity. Qur'an QA (Malhas et al., 2022, 2023) established benchmarks for comprehension and passage retrieval, while (Aleid and Azmi, 2025) supports research on fatwa related to Hajj (Muslim pilgrimage). Most approaches mitigate hallucinations through retrieval-augmented generation (RAG) (Lewis et al., 2020), conservative prompting, and reranking rather than explicit detectors. Recent frameworks such as EMAN (El Ganadi et al., 2025) stress governance and cultural alignment when deploying LLMs on Islamic texts.

Overall, prior work shows progress but also gaps: (i) reliance on mitigation rather than calibrated detectors in high-stakes religious contexts, and (ii) lack of standardized evaluation for detecting misquotations or unsupported doctrinal claims. Our work builds on these efforts by extending hallucination detection to Arabic religious texts with domain-grounded and span-level evaluation.

A.2 Qur'an QA 2022 and 2023

With Qur'an and Hadith QA being a continuation of Qur'an QA 2022[12] (Malhas et al., 2022) and Qur'an QA 2023[13] (Malhas et al., 2023) shared tasks, we provide an overview of those two editions.

The Qur'an QA shared task in its first round (2022) comprised a single machine reading comprehension (MRC) task over the Holy Qur'an: given a passage of consecutive verses from one Surah and an MSA question about that passage, systems had to extract *any* correct answer *span*. The main measure used in evaluation was partial Reciprocal Rank (pRR) (Malhas and Elsayed, 2020). The task attracted 30 teams, 13 of which submitted 30 runs in the test phase. Ten system description papers were published in OSACT 2022 (Al-Khalifa et al., 2022), and the best-performing systems achieved pRR=0.586, underscoring the difficulty of the MRC task. Leading systems (ElKomy and Sarhan, 2022; Ahmed et al., 2022) mainly used fine-tuned encoder-only BERT-based models, notably AraELECTRA (Antoun et al., 2021) and AraBERT (Antoun et al., 2020).

Qur'an QA 2023 introduced a more challenging MRC task and a new Qur'anic Passage Retrieval (QPR) task, which parallels the Qur'an QA component of Subtask 2 in the present shared task. The primary goal of QPR is to retrieve *all* Qur'anic passages that contain potential answers to a question posed in MSA. A total of 38 and 29 teams registered for QPR and MRC, respectively, and 10 teams submitted 39 runs in the test phase across the two tasks. The evaluation results revealed the inherent difficulty of the tasks: the top team achieved $pRR = 0.571$ on MRC and $MAP@10 = 0.251$ on QPR. For MRC, fine-tuned AraELECTRA and AraBERT models remained leading performers for the top team that employed them. Notably, the second-place team was the only one to adopt a GPT-4 model in a zero-shot prompt setting (Zekiye and Amroush, 2023). For QPC, the top-performing approach ensembled dual- and cross-encoder BERT-based models with staged fine-tuning on Arabic QA and domain-specific datasets (Elkomy and Sarhan, 2023). Attempts to use LLMs as embedding models or re-rankers were modest and did not feature among the top systems.

[12] https://sites.google.com/view/quran-qa-2022

[13] https://gitlab.com/bigirqu/quran-qa-2023

B Prompt for Generating Responses with Qur'anic and Hadith Evidence

Prompt:

أجب عن السؤال التالي واستشهد بآيات من القرآن الكريم وأحاديث شريفة.

السؤال:

Translation:

Provide an answer to the following question, citing evidence from the Qur'an and Prophetic Hadiths.
Question:

C Annotation and Correction Guidelines

1. Incomplete texts: Any incomplete Qur'anic verse (Ayah) or incomplete Hadith is considered an error.

2. Diacritization: Incorrect diacritization is marked as an error, whereas partially correct diacritization or the absence of diacritics is not treated as an error.

3. Error granularity: A single error in an Ayah or Hadith suffices to label the span as erroneous.

4. Reference verification: In this version, verification of metadata such as chapter or Hadith reference numbers is not required.

5. Span boundaries: Annotated spans exclude outer punctuation marks, if present.

6. Multiple Ayahs: If more than one Ayah appears in the same span, the entire span is selected even if an internal verse number appears in the middle.

7. Sources: Corrected Qur'anic text must be copied from `https://quran.ksu.edu.sa/`, and corrected Hadith from `https://dorar.net/hadith`.

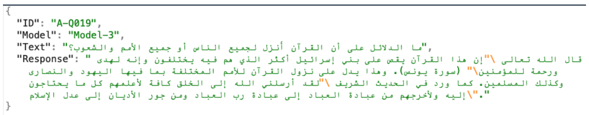
8. Correction task: Annotators predict the intended Ayah or Hadith and copy the exact corrected and complete text from the designated sources. If no valid correction can be determined, they write "Wrong" in the correction field.

9. Consistency in length: Corrections must preserve the number of intended Ayahs. For instance, if the erroneous text contains two Ayahs, the corrected version should also contain two.

10. Output assessment: In this version, we focus solely on the verification and correction of Qur'anic verses (Ayahs) and Hadiths, without assessing the completeness of the answers or their relevance to the given question. We leave this for future releases.

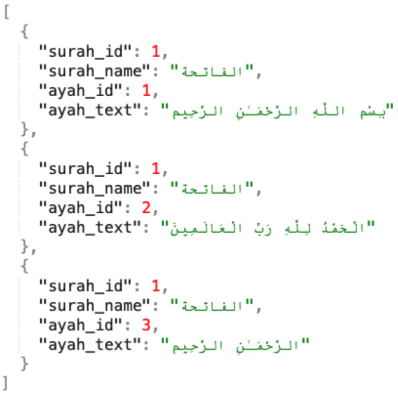
11. Correction formatting: For quality control, annotators were instructed to prepend a serial number to each correction in the text area, reflecting its order in the list of erroneous Ayahs or Hadiths. Each correction was required to be written on a separate line.

D Sample Data Files Provided to Participants



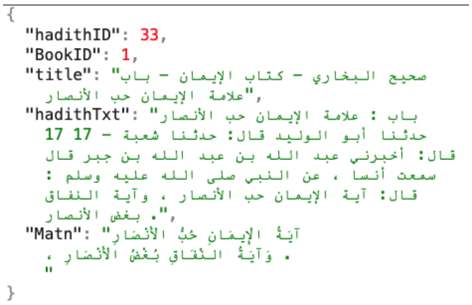
```
{
  "ID": "A-Q019",
  "Model": "Model-3",
  "Text": "ما الدلائل على أن القرآن أنزل لجميع الناس أو جميع الأمم والشعوب؟",
  "Response": " قال الله تعالى \"إن هذا القرآن يقص على بني إسرائيل أكثر الذي هم فيه يختلفون وإنه لهدى
    ورحمة للمؤمنين\" (سورة يونس). وهذا يدل على نزول القرآن للأمم المختلفة بما فيها اليهود والنصارى
    وكذلك المسلمين. كما ورد في الحديث الشريف \"لقد أرسلني الله إلى الخلق كافة لأعلمهم كل ما يحتاجون
    إليه ولأخرجهم من عبادة العباد إلى عبادة رب العباد ومن جور الأديان إلى عدل الإسلام\"."
}
```

Figure 4: Example entry from the development set showing a question, model ID, and model-generated response.



```
[
  {
    "surah_id": 1,
    "surah_name": "الفاتحة",
    "ayah_id": 1,
    "ayah_text": "بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ"
  },
  {
    "surah_id": 1,
    "surah_name": "الفاتحة",
    "ayah_id": 2,
    "ayah_text": "الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ"
  },
  {
    "surah_id": 1,
    "surah_name": "الفاتحة",
    "ayah_id": 3,
    "ayah_text": "الرَّحْمَٰنِ الرَّحِيمِ"
  }
]
```

Figure 5: Sample JSON entries from the Qur'an reference collection, showing Surah name, Ayah ID, and Ayah text.



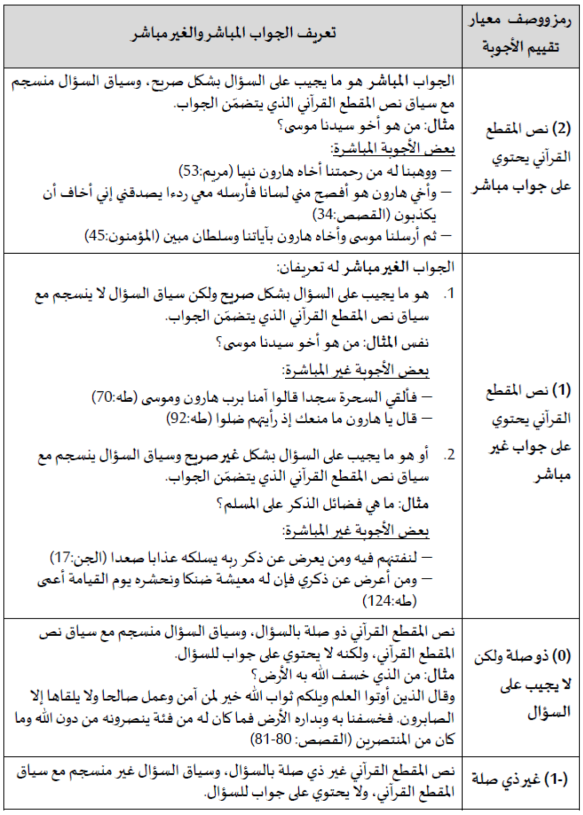
```
{
  "hadithID": 33,
  "BookID": 1,
  "title": "صحيح البخاري - كتاب الإيمان - باب
    علامة الإيمان حب الأنصار",
  "hadithTxt": "باب : علامة الإيمان حب الأنصار
    حدثنا أبو الوليد قال: حدثنا شعبة - 17 17
    قال: أخبرني عبد الله بن عبد الله بن جبر قال
    سمعت أنسا ، عن النبي صلى الله عليه وسلم :
    قال: آية الإيمان حب الأنصار ، وآية النفاق
    بغض الأنصار .",
  "Matn": "آيَةُ الإِيمَانِ حُبُّ الأَنْصَارِ
    . وَآيَةُ النِّفَاقِ بُغْضُ الأَنْصَارِ ،
    "
}
```

Figure 6: Sample JSON entry from the Hadith reference collection, including metadata (Book ID, title) and Hadith text.

E Annotation Rubrics for Qur'an Hadith QA Subtask

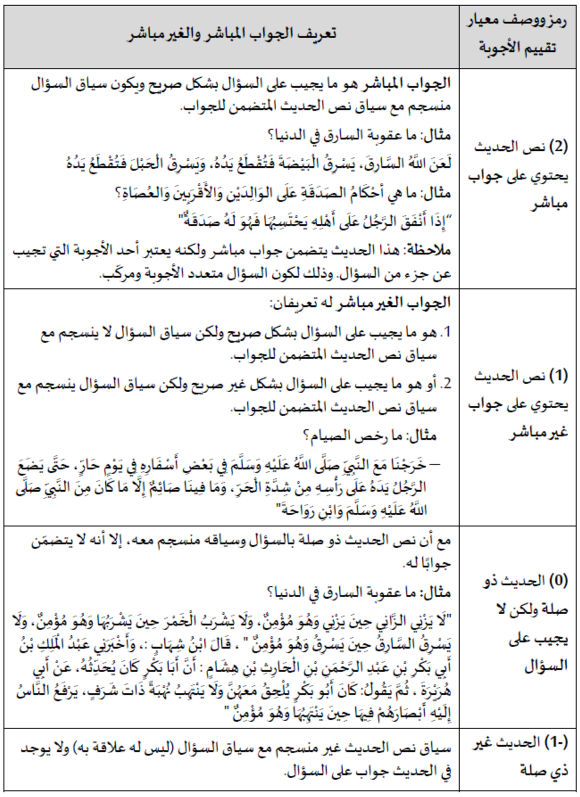
Team Name	Run	MAP@10	MAP_Q@5	MAP_H@5
Burhan	351588_Burhan_PQQFHF	0.3351	0.3389	0.3876
Burhan	351587_Burhan_QFHF	0.3021	0.3091	0.3461
Burhan	351586_Burhan_QFH	0.2916	0.3130	0.2936
BurhanAI	351568_burhanai_task_2_RAG_gpt5high	0.2807	0.3257	0.2386
ThinkDrill	351792_run_sample	0.2296	0.2623	0.2150
NUR	351549_nur_run01	0.1809	0.2334	0.1923
NUR	351550_nur_run02	0.1804	0.2257	0.1961
BayaNet	351272_BayaNet_run02mod	0.1504	0.2064	0.2240
NUR	351551_nur_run03	0.1257	0.1438	0.1569
MSA	350916_MSA_02	0.1185	0.1674	0.0685
MSA	351316_MSA_04	0.1185	0.1674	0.0685
MSA	351275_MSA_03	0.1185	0.1674	0.0685
ThinkDrill	351585_run_sample	0.0509	0.0977	0.0841
Maged	351633_run_sample	0.0332	0.0887	0.0457
ThinkDrill	351580_run_sample	0.0226	0.0482	0.1569
BayaNet	351263_BayaNet_b6453eb4	0.0157	0.0205	0.0067
CISRG	350176_CISRG_r25	0.0116	0.0294	0.0128
Maged	351629_run_sample	0.0000	0.1569	0.1961
Maged	351462_run_sample	0.0000	0.0588	0.0196

Table 4: The evaluation results of the last three runs submitted to Subtask 2 ranked by MAP@10. Teams with * did not submit a system paper. The run name is formatted as CodaBenchSubmissionID_RunName.



رمزووصف معيار تقييم الأجوبة	تعريف الجواب المباشر والغير مباشر
(2) نص المقطع القرآني يحتوي على جواب مباشر	الجواب المباشر هو ما يجيب على السؤال بشكل صريح، وسياق السؤال منسجم مع سياق نص المقطع القرآني الذي يتضمَن الجواب. مثال: من هو أخو سيدنا موسى؟ بعض الأجوبة المباشرة: – ووهبنا له من رحمتنا أخاه هارون نبيا (مريم:53) – وأخي هارون هو أفصح مني لسانا فأرسله معي ردءا يصدقني إني أخاف أن يكذبون (القصص:34) – ثم أرسلنا موسى وأخاه هارون بآياتنا وسلطان مبين (المؤمنون:45)
(1) نص المقطع القرآني يحتوي على جواب غير مباشر	الجواب الغير مباشر له تعريفان: 1. هو ما يجيب على السؤال بشكل صريح ولكن سياق السؤال لا ينسجم مع سياق نص المقطع القرآني الذي يتضمَن الجواب. نفس المثال: من هو أخو سيدنا موسى؟ بعض الأجوبة غير المباشرة: – فألقي السحرة سجدا قالوا آمنا برب هارون وموسى (طه:70) – قال يا هارون ما منعك إذ رأيتهم ضلوا (طه:92) 2. أو هو ما يجيب على السؤال بشكل غير صريح وسياق السؤال ينسجم مع سياق نص المقطع القرآني الذي يتضمَن الجواب. مثال: ما هي فضائل الذكر على المسلم؟ بعض الأجوبة غير المباشرة: – لنفتنهم فيه ومن يعرض عن ذكر ربه يسلكه عذابا صعدا (الجن:17) – ومن أعرض عن ذكري فإن له معيشة ضنكا ونحشره يوم القيامة أعمى (طه:124)
(0) ذو صلة ولكن لا يجيب على السؤال	نص المقطع القرآني ذو صلة بالسؤال، وسياق السؤال منسجم مع سياق نص المقطع القرآني، ولكنه لا يحتوي على جواب للسؤال. مثال: من الذي خسف الله به الأرض؟ وقال الذين أوتوا العلم ويلكم ثواب الله خير لمن آمن وعمل صالحا ولا يلقاها إلا الصابرون. فخسفنا به وبداره الأرض فما كان له من فئة ينصرونه من دون الله وما كان من المنتصرين (القصص: 80-81)
(-1) غير ذي صلة	نص المقطع القرآني غير ذي صلة بالسؤال، وسياق السؤال غير منسجم مع سياق المقطع القرآني، ولا يحتوي على جواب للسؤال.

Figure 7: Rubric for annotating potential answer-bearing Qur'anic passages to a given question.



رمزووصف معيار تقييم الأجوبة	تعريف الجواب المباشر والغير مباشر
(2) نص الحديث يحتوي على جواب مباشر	الجواب المباشر هو ما يجيب على السؤال بشكل صريح ويكون سياق السؤال منسجم مع سياق نص الحديث المتضمن للجواب. مثال: ما عقوبة السارق في الدنيا؟ لَعَنَ اللَّهُ السَّارِقَ، يَسْرِقُ الْبَيْضَةَ فَتُقْطَعُ يَدُهُ، وَيَسْرِقُ الْحَبْلَ فَتُقْطَعُ يَدُهُ مثال: ما هي أَحْكَامُ الصَدَقَةِ عَلَى الوَالِدَيْنِ وَالأَقْرَبِينَ وَالعُصَاةِ؟ "إِذَا أَنْفَقَ الرَّجُلُ عَلَى أَهْلِهِ يَحْتَسِبُهَا فَهُوَ لَهُ صَدَقَةٌ" ملاحظة: هذا الحديث يتضمن جواب مباشر ولكنه يعتبر أحد الأجوبة التي تجيب عن جزء من السؤال. وذلك لكون السؤال متعدد الأجوبة ومركَب.
(1) نص الحديث يحتوي على جواب غير مباشر	الجواب الغير مباشر له تعريفان: 1. هو ما يجيب على السؤال بشكل صريح ولكن سياق السؤال لا ينسجم مع سياق نص الحديث المتضمن للجواب. 2. أو هو ما يجيب على السؤال بشكل غير صريح ولكن سياق السؤال ينسجم مع سياق نص الحديث المتضمن للجواب. مثال: ما رخص الصيام؟ – خَرَجْنَا مَعَ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ فِي بَعْضِ أَسْفَارِهِ فِي يَوْمٍ حَارٍّ، حَتَّى يَضَعَ الرَّجُلُ يَدَهُ عَلَى رَأْسِهِ مِنْ شِدَّةِ الْحَرِّ، وَمَا فِينَا صَائِمٌ إِلَّا مَا كَانَ مِنَ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ وَابْنِ رَوَاحَةَ"
(0) الحديث ذو صلة ولكن لا يجيب على السؤال	مع أن نص الحديث ذو صلة بالسؤال وسياقه منسجم معه، إلا أنه لا يتضمَن جوابًا له. مثال: ما عقوبة السارق في الدنيا؟ "لَا يَزْنِي الزَّانِي حِينَ يَزْنِي وَهُوَ مُؤْمِنٌ، وَلَا يَشْرَبُ الْخَمْرَ حِينَ يَشْرَبُهَا وَهُوَ مُؤْمِنٌ، وَلَا يَسْرِقُ السَّارِقُ حِينَ يَسْرِقُ وَهُوَ مُؤْمِنٌ " . قَالَ ابْنُ شِهَابٍ :. وَأَخْبَرَنِي عَبْدُ الْمَلِكِ بْنُ أَبِي بَكْرِ بْنِ عَبْدِ الرَّحْمَنِ بْنِ الْحَارِثِ بْنِ هِشَامٍ : أَنَّ أَبَا بَكْرٍ كَانَ يُحَدِّثُهُ، عَنْ أَبِي هُرَيْرَةَ ، ثُمَّ يَقُولُ: كَانَ أَبُو بَكْرٍ يُلْحِقُ مَعَهُنَّ وَلَا يَنْتَهِبُ نُهْبَةً ذَاتَ شَرَفٍ، يَرْفَعُ النَّاسُ إِلَيْهِ أَبْصَارَهُمْ فِيهَا حِينَ يَنْتَهِبُهَا وَهُوَ مُؤْمِنٌ "
(-1) الحديث غير ذي صلة	سياق نص الحديث غير منسجم مع سياق السؤال (ليس له علاقة به) ولا يوجد في الحديث جواب على السؤال.

Figure 8: Rubric for annotating potential answer-bearing Hadith *matns* to a given question.

NUR at IslamicEval 2025 Shared Task: Retrieval-Augmented LLMs for Qur’an and Hadith QA

Serag Amin[1]* **Ranwa Aly**[1]* **Yara Allam**[1]*
Yomna Eid[2] **Ensaf H. Mohamed**[2]
[1] Faculty of Computers and Data Science, Alexandria University
[2] Center for Informatics Science (CIS),
School of Information Technology and Computer Science, Nile University
cds.{seragamin23144,ranwakhaled30408,yaraibrahim23394}@alexu.edu.eg
{YEid,EnMohamed}@nu.edu.eg

Abstract

In this paper, we present our contribution to the IslamicEval 2025 shared task. More specifically, we address subtask 2, which is a passage retrieval (PR) system for Qur’an and Hadith, the two central bodies of text in Islam. Basing off of a fine-tuned BERT-based sentence transformer retrieval model, we explore several approaches, including pipelined fine-tuning of cross-encoders, as well as using a state-of-the-art LLM for reranking of relevant passages, and identification of zero-answer questions. Our best-performing system achieves a MAP@10 of 0.1809, MAP_Q@5 of 0.2334, and MAP_H@5 of 0.1923 on the test set.

1 Introduction

As the two primary sources for Islamic teachings, the Holy Qur’an and the Hadith are essential to the lives of roughly 2 billion Muslims. They contain rulings, moral and spiritual guidance, and general ways of life, making Islamic question answering (QA) systems extremely important for those practising, and even inquisitive non-Muslims. It is also important that such systems maintain high accuracy and reliability, as small errors or hallucinations may have significant implications due to the sensitivity of the materials.

While QA in Arabic has been tackled previously (Koto et al., 2024) and remains an active research area, the challenge with Arabic morphological richness is amplified even more when it comes to religious texts, where context, syntax, or vocabulary can change a passage’s meaning entirely. Previously, the Qur’an QA 2022 (Malhas et al., 2022) and Qur’an QA 2023 (Malhas et al., 2023) shared tasks addressed this challenge, but only within the scope of the Holy Qur’an. In comparison, Hadith collections present a broader, more complex challenge for information retrieval (IR). Hadith is built upon a chain of narrators quoting the Prophet Muhammad, peace be upon him, varying in length, phrasing, and authenticity, and spread across multiple compilations. They also lack a unified indexing system, as opposed to the Qur’an, which constitutes a singular source of information. This leads to a more dynamic and realistic approach in the IslamicEval shared task (Mubarak et al., 2025). For a free-text question in Modern Standard Arabic (MSA), the system must retrieve a ranked list of up to 20 Qur’anic passages or Hadiths that may contain the answer to the question. The question could also be unanswerable. In some cases, the question may also have no relevant answer in the Qur’an but one in the Hadith, or vice versa, requiring systems to be versatile in searching across both corpora.

Similar to the previous editions, the task provides us with a set of thematic Qur’an passages, as well as the Sahih Al-Bukhari Hadith collection. We are also provided with the *AyaTEC* Qur’an QA dataset (Malhas and Elsayed, 2020), as discussed further in Section 2. However, no equivalent exists for Hadith QA, prompting us to search for relevant external sources for training our systems.

Our contribution to the subtask involves pipelined fine-tuning of BERT-based sentence transformer models for the retrieval of relevant documents, followed by either a fine-tuned cross-encoder or a state-of-the-art LLM for filtering and identification of zero-answer questions. The system is then evaluated on mean average precision, specifically, MAP@10 and MAP@5 for the Qur’an and Hadith passages independently. The paper is structured as follows: Section 2 describes the data used for our experiments, Section 3 goes into the details of the experiments and provides an overview of the results achieved, and Section 5 discussing and drawing insights from these results. Lastly, Section 6 offers a conclusion to our work. We release our code and data publicly on GitHub[1].

*These authors contributed equally to this work.

[1] https://github.com/Yoriis/IslamicEval2025

Proceedings of The Third Arabic Natural Language Processing Conference, pages 494–502
November 8-9, 2025 ©2025 Association for Computational Linguistics

Split	Train	Dev	Test
# Question–passage pairs	1261	298	–
# Questions			
Multi-answer	131 (62%)	26 (65%)	–
Single-answer	48 (23%)	8 (20%)	–
Zero-answer	31 (15%)	6 (15%)	–
Total	**210**	**40**	**71**

Table 1: AyaTEC v1.3 Split Distribution

2 Data

The task data consisted of three parts: the Thematic Qur'anic Passage collection (QPC) (Swar, 2007), containing *1,266* thematic passages that cover the whole Holy Qur'an in a simple-clean text style, without diacritics, the Sahih Al-Bukhari Hadith collection (Al-Sharjy and Al-Zubaidi, 2009), comprising *2,254* Hadiths, the authors having excluded redundant Hadiths and Arabic commentary, and the *AyaTEC v1.3* (Malhas and Elsayed, 2020) dataset, composed of question-passage pairs. A brief description of the split can be found in Table 1.

As the dataset size remains limited, we adopt a sequential fine-tuning strategy, adding increasingly task-specific datasets to enhance the model's adaptation to our domain. We use the Arabic portion of the TyDi dataset (Clark et al., 2020), containing about 15 thousand QA pairs. We use the Jalalayn Tafseer of the Qur'an, aggregated to the thematic passages provided. Additionally, to address the limited size of task-specific data, especially for Hadith, we use the QuQA and HaQA datasets (Alnefaie et al., 2023), which contain *3382* and *1598* QA pairs, respectively. Lastly, to increase models' sensitivity to zero-answer questions, we augment each of our datasets with several random negative samples - 5 negatives per sample for HaQA, and 3 negatives per sample for the others.

3 System

Our system has 2 stages: retrieval & re-ranking, discussed in this section & illustrated in figure 1.

3.1 Retrieval

To retrieve the top-K passages for a question, we encode the question and all thematic Qur'anic passages and Hadiths using a sentence embedding model, compute cosine similarity, and rank the passages. We evaluated several Arabic embedding models on the shared task's Qur'an-only dev set using **Recall@30** to establish baseline performance; results are in Table 10. The best model, `AraModernBert`[2], achieved a Recall@30 of **0.445**.

Retriever Fine-Tuning Starting from the AraModernBERT model, we fine-tuned for the target domains using the shared task's Qur'an-only training set and additional data (Section 2). Following prior work on dense retrieval with hard negatives (Zhan et al., 2021; ElKomy and Sarhan, 2023), we retrieved top-ranked Qur'anic and Hadith passages per question using the base model for fine-tuning. For each query, we sampled K passages in total, including multiple positives and treating the rest as hard negatives. We also tested positive-only fine-tuning to assess the impact of excluding negatives.

We implemented two fine-tuning pipelines, each using both cosine and contrastive loss:

- **Pipeline A**: Single-stage fine-tuning on the shared task's Qur'an-only training data.
- **Pipeline B**: Multi-stage curriculum fine-tuning using additional QA datasets (Section 2), starting with TyDiQA, followed by Tafseer, QuQA, HaQA, and finally the Qur'an-only set.

For both pipelines, we varied the number of *passages* (K) used during fine-tuning. Each K includes multiple positive and hard negative passages, retrieved from both Qur'an and Hadith corpora. We evaluated performance using recall at multiple retrieval depths, excluding unanswerable questions.

Pipeline A Direct fine-tuning (Table 2) shows strong gains over the positive-only baseline, with its best Recall@30 at **0.491** exceeding the baseline by over 20 percent. At larger retrieval depths, Recall@70 peaks at **0.592**.

Passages	Loss	Search	R@30	R@50	R@70
Positive only	Contrastive	Cosine	0.285	0.329	0.385
50	Contrastive	Cosine	0.462	**0.537**	0.555
70	Cosine	Cosine	0.446	**0.537**	**0.592**
	Contrastive	L2	**0.491**	0.521	0.552

Table 2: Top-performing configurations in Pipeline A by number of passages. Full results in Table 11.

Pipeline B The multi-stage curriculum (Table 3) surpasses Pipeline A at both shallow and deep retrieval. Its best Recall@30 reaches **0.541**, around 5 percent higher than Pipeline A, while its Recall@70 climbs to **0.688**, nearly 10 percent above Pipeline A's top result.

[2] `https://huggingface.co/NAMAA-Space/AraModernBert-Base-STS`

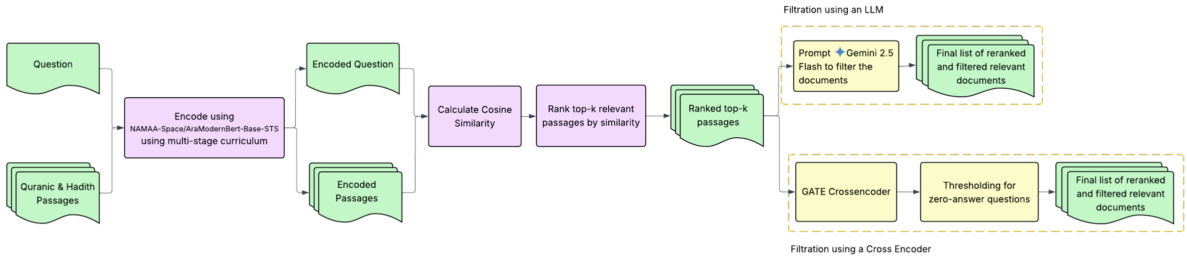

Filtration using an LLM
Question
Quranic & Hadith Passages
Encode using NAMAA-Space/AraModernBert-Base-STS using multi-stage curriculum
Encoded Question
Encoded Passages
Calculate Cosine Similarity
Rank top-k relevant passages by similarity
Ranked top-k passages
Prompt Gemini 2.5 Flash to filter the documents
Final list of reranked and filtered relevant documents
GATE Crossencoder
Thresholding for zero-answer questions
Final list of reranked and filtered relevant documents
Filtration using a Cross Encoder

Figure 1: Figure showing the final pipeline used for the submitted runs. **Green** shapes represent input and output data modules, **purple** boxes denote retrieval processes, and **yellow** boxes signify reranking and filtering stages.

Passages	Loss	Search	R@30	R@50	R@70
60	Cosine	L2	0.505	0.600	**0.688**
70	Contrastive	Cosine	**0.541**	0.581	0.640
80	Contrastive	L2	0.537	**0.620**	0.645

Table 3: Top-performing configurations in Pipeline B by number of passages. Full results in Table 12

3.2 Reranking

To re-rank the retrieved documents, we experimented with two approaches: a cross-encoder architecture and a large language model.

3.2.1 Cross-Encoder Architecture

Building on the fine-tuned retrieval model, we use **Pipeline B** to fine-tune two cross-encoders: `AraBERTv0.2-base` (Antoun et al., 2020), and `NAMAA Space GATE Reranker V1` (GATE) (NAMAA-Space, 2025). Our choice of models is guided by the Arabic RAG leaderboard (Mohaned A. Rashad, 2025), which evaluates retrieval and reranking systems. GATE, built on `AraBERT` and `Arabic Triplet Matryoshka` (Nacar et al., 2025), ranks highly on this benchmark while also remaining resource-efficient. `AraBERTv0.2-base`, as one of the earliest widely adopted Arabic Transformers and GATE's predecessor, serves as a baseline for comparison. For identification of zero-answer questions, we use a thresholding-based approach. If all passages, after reranking, have scores below the threshold, the question is deemed to have no answers, and the systems returns -1.

Two versions of **Pipeline B** were experimented with. In one configuration, we drop the Tafseer dataset for fine-tuning, and exclude the task data as well **(Pipeline B1)**. This generally led to better results, as seen in Table 4. In the other, we utilize the full pipeline, ending with fine-tuning independently on two versions of the task data: one with only positive passages sampling, and one with Top-70 **(Pipeline B2)**. A representation of both pipelines can be found in Figure 2. Table 5 shows the **MAP@5** and **MAP@10** for the dev set after each fine-tuning step. Interestingly, in both scenarios, fine-tuning on the task data decreases performance.

It's also important to note that a k-value of *70* was used to retrieve the relevant passages, which were then reranked, and the scoring threshold for zero-answer questions was set at *0.15* for these experiments. We experimented with the thresholding hyperparameter, as can be seen in Appendix C.

Model	Metric	Baseline	TYDI	QUQA	HAQA
GATE	MAP@5	**0.3172**	0.2319	0.2372	0.2548
	MAP@10	**0.3215**	0.2503	0.2574	0.2786
AraBERT	MAP@5	0.0278	0.1712	0.1965	**0.2186**
	MAP@10	0.0371	0.1972	0.2138	**0.2334**

Table 4: MAP@5 and MAP@10 scores without Tafseer

The regression in GATE's performance could be attributed to several factors. This model has already been pretrained on large-scale Arabic corpora, and further fine-tuning likely introduced overfitting and reduced the model's ability to generalize. Additionally, the negative sampling strategies may not have been comprehensive enough to evaluate the reranker's ability to improve from the baseline. This suggests that, for already high-performing rerankers, there's a need for more careful design of fine-tuning data, otherwise it might be better to use the reranker without further training.

3.2.2 LLM-based Approach

We used Gemini 2.5 Flash (Comanici et al., 2025) to rerank retrieved documents by instructing it to get an ordered list of passage IDs that have the answers to a given question according to their relevance. The prompt design process included adding more instructions about the format of the answers to avoid hallucination of passages and emphasizing the importance of relevance and order of the re-

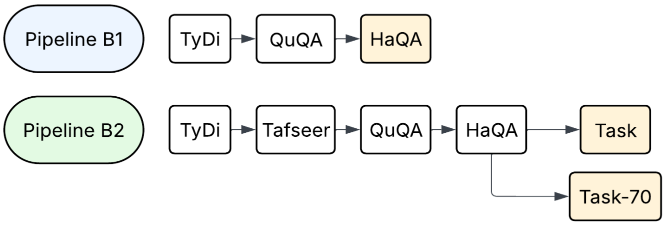


Figure 2: Figure showing cross-encoder finetuning configurations.

Model	Metric	Baseline	TyDi	Tafseer	QuQA	HaQA	Task	Task-70
GATE	MAP@5	**0.3172**	0.2319	0.2504	0.2318	0.2499	0.2107	0.2480
	MAP@10	**0.3215**	0.2503	0.2642	0.2563	0.2736	0.2367	0.2680
AraBERT	MAP@5	0.0278	0.1712	0.2099	0.1899	0.1884	0.1733	**0.2039**
	MAP@10	0.0371	0.1972	0.2099	0.2093	0.2081	0.1967	**0.2267**

Table 5: MAP@5 and MAP@10 cross-encoder scores on the dev set for full Pipeline B

turned passage IDs. The final prompt used is found in Appendix A.

Experimentation with different k values showed that higher values produced inconsistent results with Gemini, with MAP ranges varying drastically (Table 6). However, Gemini showed relatively reliable performance with the top 70 passages to filter across runs and models.

Pre and Post Retrieval Enhancements: To improve the performance of our pipeline, we experimented with two approaches: one for *pre-retrieval* and one for *post-retrieval*.

Our proposed technique for **pre-retrieval** is to use *topic filtering* before passing the question to our RAG model. This method uses Latent Dirichlet Allocation (LDA) to find the topics in the reranking stage (Ampazis, 2024). We applied it as a pre-retrieval technique by assigning, using the LLM, each question and Qur'anic passage a list of one or more topics out of 40 relevant topics in Islam, found in Appendix B. The filtering reduced the search space for the RAG model by providing it only with the documents matching the topics in the question to encode. Results in Table 7 show that performance increases without topic filtering, with MAP improving by **2%+**.

For **post-retrieval**, to enhance the LLM's understanding of the retrieved Qur'anic passages, we expanded each passage with its interpretation by aggregating the Jalalayn Tafseer. We observe that adding Tafseer reduced performance, as Gemini struggles with longer inputs, yielding at best MAP@10 of **0.15**.

Model Name	Top K	MAP@5	MAP@10
Baseline Model	70	0.2983	0.3137
	80	**0.3048**	**0.3294**
	100	0.2552	0.2902
Pipeline A	70	**0.3311**	**0.3579**
	80	0.2777	0.3049
	100	0.2913	0.3185
Pipeline B	70	0.3506	0.3801
	80	0.3550	0.3550
	100	**0.3598**	**0.3888**

Table 6: MAP@5 and MAP@10 scores for different models across varying Top K values.

Model Name	Top K	MAP@5	MAP@10
With Topic Modeling	30	0.3991	0.4299
	70	0.3958	0.4365
Without Topic Modeling	30	0.4228	0.4491
	70	**0.4407**	**0.4591**

Table 7: MAP@5 and MAP@10 scores for different models across varying Top K values.

4 Results

For evaluation on the test set, we chose three configurations: the first two use Gemini, with the first retrieving the top 70 most relevant documents from the combined collection Qur'an and Hadith passages, and the second retrieving 50 from Qur'an and 20 from Hadith to allow for higher representation of Hadith. The last approach also followed this method with the fine-tuned GATE model **(Pipeline B1)** used for filtering. It's important to note that when retrieving Hadith passages, we removed the diacritics from the texts. Results can be seen in

Table 8.

Model	MAP@10	MAP_Q@5	MAP_H@5
Gemini	**0.1809**	**0.2334**	0.1923
Gemini (50-20)	0.1804	0.2257	**0.1961**
GATE (50-20)	0.1257	0.1438	0.1569

Table 8: Subtask 2 Test Set Results

Gemini achieved higher performance than GATE in both configurations, with improvements observed across all metrics. However, in all three test runs, we observe a consistent and significant drop in performance compared to the development set.

This decline may be attributed to domain shift between the Qur'an-only development set and the mixed-source test set, or to overfitting on the fine-tuning data. While reranking with Gemini improved overall relevance, its performance on previously unseen questions proved less stable. GATE, although more consistent, remained behind Gemini, likely due to its limited capacity to model question semantics compared to the LLM-based reranker.

5 Discussion

Our experiments on the retrieval model reveal 3 key insights. First, **positive-only fine-tuning consistently underperformed** compared to using hard negatives, as both cosine and contrastive losses benefit from distinguishing relevant from highly similar but irrelevant passages. Second, the **optimal top-K passages for positive and hard negative sampling** was typically **60–80 passages**; larger values often introduced easy negatives that weakened learning. Third, there was **no single best loss–search pairing**, with outcomes varying across settings. Finally, multi-stage curriculum (Pipeline B) consistently outperformed direct fine-tuning (Pipeline A), with **up to a 10% recall improvement** at higher Recall@K values. This demonstrates the advantage of gradual domain adaptation, moving from general Arabic QA to Qur'anic and Hadith retrieval, which helps the model capture the linguistic and semantic characteristics. For filtering, Gemini had a better performance; its understanding of the passages led to an increase of more than **5%** in MAP compared to the cross-encoder results. However, adding more context - whether by increasing the number of retrieved documents or by adding Tafseer - resulted in a substantial drop in scores.

6 Conclusion

In this study, we explore QA techniques for subtask B of the IslamicEval 2025 shared task. We compare direct fine-tuning and a multi-stage approach for retrieval, and a cross-encoder and LLM for reranking. Our experiments led to an increase in Recall for retrieval and MAP for reranking compared to prior models, demonstrating the potential of our approach for building more accurate and reliable Islamic QA systems.

Limitations The main challenge is dataset size and a lack of Hadith QA pairs. Additionally, Gemini fluctuated and produced inconsistent scores across runs. GPU limitations also prevented us from carrying out experiments using larger models. The limited timeline of our experiments also prevented us from exhausting all possible configurations, hyperparameters, and other approaches.

References

A. B. A Al-Sharjy and Z. Al-Zubaidi. 2009. *Al-Tajreed Al-Sareeh of Collective Sahih Hadith.* .

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Haqa and quqa: Constructing two arabic question-answering corpora for the quran and hadith. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 90–97.

Nikolaos Ampazis. 2024. Improving RAG quality for large language models with topic-enhanced reranking. In *Proceedings of the 2024 IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI 2024)*, pages 74–87, Cham. Springer Nature Switzerland.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, and et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities.

Mohammed ElKomy and Amany Sarhan. 2023. Tce at qur'an qa 2023 shared task: Low resource enhanced transformer-based ensemble approach for qur'anic qa. In *Proceedings of the Qur'an QA 2023 Shared Task*. Tanta University, Egypt.

Fajri Koto, Haonan Li, Sara Shatanawi, Jad Doughman, Abdelrahman Boda Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. Arabicmmlu: Assessing massive multitask language understanding in arabic. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Rana Malhas and Tamer Elsayed. 2020. Ayatec: Building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(6).

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 *Proceedinsg 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.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Hamza Shahid Mohaned A. Rashad. 2025. The arabic rag leaderboard. urlhttps://huggingface.co/spaces/Navid-AI/The-Arabic-Rag-Leaderboard.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Omer Nacar and Anis Koubaa. 2024. Enhancing semantic similarity understanding in arabic nlp with nested embedding learning.

Omer Nacar, Anis Koubaa, Serry Sibaee, 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.

NAMAA-Space. 2025. Gate-reranker-v1. `https://huggingface.co/NAMAA-Space/GATE-Reranker-V1`. Hugging Face model, Apache-2.0 license. Accessed: 15 August 2025.

Marwan N. Swar. 2007. *Mushaf Al-Tafseel Al Mawdoo'ee*. Dar Al-Fajr Al-Islami, Damascus.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 1503–1512, New York, NY, USA. Association for Computing Machinery.

A Prompting

The following prompt was used for filtering and reranking using Gemini2.5 Flash:

Given a question in Modern Standard Arabic (MSA) and a list of Quranic and Hadith verses (each with an associated ID), identify the IDs of the verses that contain the answer to the question.
Instructions:
*- Return only the **IDs** of the extremely relevant verses in a **list**, **ordered** from most relevant to least relevant.*
- Do not explain your answer or provide verse text.
*- If the answer is not found in any verse, or you are unsure, **you must return [-1]**.*
*- Use the verse ID **exactly as provided** (e.g., if the verse ID is 23:14-16, return [23:14-16]).*
Question: <QUESTION-TEXT>
Verses: <RETRIEVED-PASSAGES>

B Topic Modeling

To reduce the search space of the retrieval model, we adapted a pre-retreival topic filtering approach where we assign the questions and documents one or more of the topics from Table 9.

C Thresholding Experimentation

Using our best available model, the GATE baseline, we experimented with different values of the scoring threshold (T). Intuitively, the most optimal values lie between 0.10 - 0.20 as can be seen in Figure 3.

Topics	التوحيد، أسماء الله وصفاته، الملائكة، القدر، اليوم الآخر، الطهارة، الصلاة، الصيام، الزكاة، الحج والعمرة، الأذكار والدعاء، الزواج، الطلاق، قضايا المرأة، الميراث، تربية الأبناء، البيع والشراء، الربا، العقود، الصدقات، التأمين، المعاملات الحديثة، الطعام والشراب، الترفيه، الأخلاق، العلاقات مع غير المسلمين، قصص الأنبياء، الرؤى والرؤية، الصيام وشهر رمضان، الزكاة والصدقات، الذكر والدعاء والرقية الشرعية، العقيدة، النبوة والسيرة النبوية، الزواج وحقوق الزوجين، الميراث والوصايا، المعاملات التجارية والمالية الحديثة، الأطعمة والأشربة والمكونات الحلال والحرام، اللباس والزينة والتجميل، المباحات، الجنايات والقضاء والسياسة الشرعية

Table 9: The list of 40 topics assigned to questions and Qur'anic passages used for filtering before retrieval.

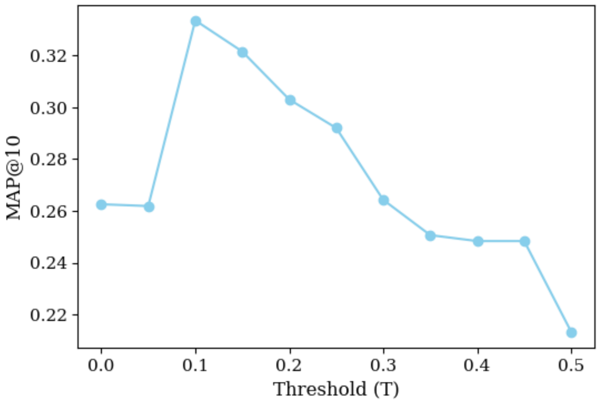


Figure 3: MAP@10 on GATE for Dev Set VS Threshold

D Arabic Embedding Model Evaluation

To identify suitable retriever models, we evaluated a broad set of Arabic (and multilingual) embedding models using cosine similarity ranking and Recall@30 on the Qur'anic development set. Due to limited computational resources, we were unable to run inference on larger-scale models (e.g., >500M parameters) with extensive batch processing, and thus prioritized models that were feasible for our hardware budget.

Model	Recall@30	Trainable Params (M)
NAMAA-Space/AraModernBert-Base-STS[1]	0.4451	149
silma-ai/silma-embeddding-sts-v0.1[2]	0.4136	135
omarelshehy/Arabic-Retrieval-v1.0[3]	0.3880	135
omarelshehy/Arabic-STS-Matryoshka-V2[4]	0.3876	135
Omartificial-Intelligence-Space/GATE-AraBert-v1(Nacar and Koubaa, 2024)	0.3663	135
ALJIACHI/bte-base-ar[5]	0.3627	149
mohamed2811/Muffakir_Embedding[6]	0.3576	135
silma-ai/silma-embeddding-matryoshka-v0.1[7]	0.3517	135
Omartificial-Intelligence-Space/Arabic-Triplet-Matryoshka-V2(Nacar and Koubaa, 2024)	0.3478	135
AhmedZaky1/arabic-bert-sts-matryoshka[8]	0.3235	135
Alibaba-NLP/gte-multilingual-base[9]	0.3073	305
Omartificial-Intelligence-Space/Arabert-all-nli-triplet-Matryoshka(Nacar and Koubaa, 2024)	0.3053	135
AhmedZaky1/arabic-bert-nli-matryoshka[10]	0.3028	135
AhmedZaky1/DIMI-embedding-v2[11]	0.2924	305
ibm-granite/granite-embedding-278m-multilingual[12]	0.2701	278
omarelshehy/arabic-english-sts-matryoshka-v2.0[13]	0.2680	560
OmarAlsaabi/e5-base-mlqa-finetuned-arabic-for-rag[14]	0.2622	278
intfloat/multilingual-e5-base(Wang et al., 2024)	0.2599	278
ibm-granite/granite-embedding-107m-multilingual[15]	0.2598	107
Abdelkareem/zaraah_jina_v3[16]	0.2443	64
AhmedZaky1/DIMI-embedding-v4[17]	0.2322	305
Snowflake/snowflake-arctic-embed-m-v2.0[18]	0.1745	305
Abdelkareem/abjd[19]	0.1677	438
Abdelkareem/ara-qwen3-18[20]	0.1677	438
Omartificial-Intelligence-Space/Arabic-labse-Matryoshka(Nacar and Koubaa, 2024)	0.1579	471
sentence-transformers/LaBSE[21]	0.1575	471
Omartificial-Intelligence-Space/Arabic-MiniLM-L12-v2-all-nli-triplet[22]	0.0973	118
mixedbread-ai/mxbai-embed-large-v1(Li and Li, 2023)	0.0357	335
metga97/Modern-EgyBert-Base[23]	0.0145	159
metga97/Modern-EgyBert-Embedding[24]	0.0145	159
sentence-transformers/all-mpnet-base-v2[25]	0.0057	109
sentence-transformers/all-MiniLM-L6-v2[26]	0.0008	23

Table 10: Recall@30 and parameter counts for reviewed sentence embedding models on the Qur'anic dev set.

[1] https://huggingface.co/NAMAA-Space/AraModernBert-Base-STS

[2] https://huggingface.co/silma-ai/silma-embedding-sts-0.1

[3] https://huggingface.co/omarelshehy/Arabic-Retrieval-v1.0

[4] https://huggingface.co/omarelshehy/Arabic-STS-Matryoshka-V2

[5] https://huggingface.co/ALJIACHI/bte-base-ar

[6] https://huggingface.co/mohamed2811/Muffakir_Embedding

[7] https://huggingface.co/silma-ai/silma-embedding-matryoshka-0.1

[8] https://huggingface.co/AhmedZaky1/arabic-bert-sts-matryoshka

[9] https://huggingface.co/Alibaba-NLP/gte-multilingual-base

[10] https://huggingface.co/AhmedZaky1/arabic-bert-nli-matryoshka

[11] https://huggingface.co/AhmedZaky1/DIMI-embedding-v2

[12] https://huggingface.co/ibm-granite/granite-embedding-278m-multilingual

[13] https://huggingface.co/omarelshehy/arabic-english-sts-matryoshka-v2.0

[14] https://huggingface.co/OmarAlsaabi/e5-base-mlqa-finetuned-arabic-for-rag

[15] https://huggingface.co/ibm-granite/granite-embedding-107m-multilingual

[16] https://huggingface.co/Abdelkareem/zaraah_jina_v3

[17] https://huggingface.co/AhmedZaky1/DIMI-embedding-v4

[18] https://huggingface.co/Snowflake/snowflake-arctic-embed-m-v2.0

[19] https://huggingface.co/Abdelkareem/abjd

[20] https://huggingface.co/Abdelkareem/ara-qwen3-18

[21] https://huggingface.co/sentence-transformers/LaBSE

[22] https://huggingface.co/Omartificial-Intelligence-Space/Arabic-MiniLM-L12-v2-all-nli-triplet

[23] https://huggingface.co/metga97/Modern-EgyBert-Base

[24] https://huggingface.co/metga97/Modern-EgyBert-Embedding

[25] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[26] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Passages	**Loss Function**	**Search Method**	**Recall@30**	**Recall@50**	**Recall@70**
Positive only	Cosine	Cosine	0.269	0.320	0.341
	Cosine	L2	0.211	0.250	0.283
	Contrastive	Cosine	0.285	0.329	0.385
	Contrastive	L2	0.242	0.288	0.346
30	Cosine	Cosine	0.417	0.480	0.537
	Cosine	L2	0.439	0.497	0.523
	Contrastive	Cosine	0.440	0.497	0.537
	Contrastive	L2	0.447	0.494	0.548
50	Cosine	Cosine	0.425	0.488	0.531
	Cosine	L2	0.431	0.476	0.501
	Contrastive	Cosine	0.462	**0.537**	0.555
	Contrastive	L2	0.457	0.536	0.555
70	Cosine	Cosine	0.446	**0.537**	**0.592**
	Cosine	L2	0.424	0.496	0.545
	Contrastive	Cosine	0.472	0.510	0.583
	Contrastive	L2	**0.491**	0.521	0.552
90	Cosine	Cosine	0.436	0.494	0.558
	Cosine	L2	0.428	0.466	0.501
	Contrastive	Cosine	0.477	0.517	0.559
	Contrastive	L2	0.460	0.518	0.555

Table 11: Performance of Fine-Tuned Configurations (Pipeline A) on Dev Set (Quran)

Passages	**Loss Function**	**Search Method**	**Recall@30**	**Recall@50**	**Recall@70**
60	Cosine	Cosine	0.508	0.586	0.675
	Cosine	L2	0.505	0.600	**0.688**
	Contrastive	Cosine	0.539	0.603	0.621
	Contrastive	L2	0.538	0.602	0.634
70	Cosine	Cosine	0.521	0.596	0.663
	Cosine	L2	0.464	0.577	0.636
	Contrastive	Cosine	**0.541**	0.581	0.640
	Contrastive	L2	0.539	0.606	0.634
80	Cosine	Cosine	0.446	0.548	0.646
	Cosine	L2	0.462	0.501	0.574
	Contrastive	Cosine	0.520	0.619	0.649
	Contrastive	L2	0.537	**0.620**	0.645

Table 12: Performance of Fine-Tuned Configurations (Pipeline B) on Dev Set (Quran)

BurhanAI at IslamicEval 2025 Shared Task: Combating Hallucinations in LLMs for Islamic Content; Evaluation, Correction, and Retrieval-Based Solution

Arij Al Adel
arij.aladel@gmail.com

Abu Bakr Soliman
abubakr@rankxy.com

Mohamed Sakher Sawan
me@sakher.co.uk

Rahaf Al-Najjar
rahaf.m.alnajjar@gmail.com

Sameh Amin
Sameh.m.amin@gmail.com

Abstract

In this paper, we describe our submission to the IslamicEval 2025 shared task, covering hallucination detection/correction and closed-world retrieval in Quranic and Hadith. We fine-tuned an LLM for detecting Quran and Hadith text spans, utilizing synthetic augmentation, diacritic variation, and morphological normalization to improve detection robustness (F1 = 87.10%) and used another reasoning model with tools (F1 = 90.06%). For validation, the accuracy is 88.60%, and for correction the accuracy is 66.56% where we employed a layered hierarchical index and search algorithm combining exact, normalized, fuzzy, and semantic matching with prompt-driven repair—to ensure canonical alignment and diacritic fidelity. For the correction stage, we also utilized a reasoning model with access to tools with an accuracy of 61.04%. Regarding the ranked answer-bearing text retrieval task, we implemented a Retrieval-Augmented Generation (RAG) system restricted to the corpora provided by the shared task, with structured output, vector-store grounding, and prompts tuned for "answer-enclosing" citations that achieve MAP@10 of 0.6199 on the development set and 0.2807 on the test set. The results highlight the value of normalization, corpus-restricted search, and reasoning models with tools in mitigating hallucinations and improving retrieval precision in low-resource religious settings and that much smaller fine-tuned models can compete with frontier models (e.g. GPT-5 high) for specialized tasks such as span detection.

1 Introduction

Despite SOTA of large language models (LLMs) in a wide range of natural language processing (NLP) tasks, they frequently hallucinate Li et al. (2024); Hikal et al. (2025); Orgad et al. (2024).

Employing Large Language Models (LLMs) to process religious texts Ganadi et al. (2025); Mohammed et al. (2025) raises different ethical concerns, which makes it a topic of special interest within the Ethics of Natural Language Processing (NLP) Hutchinson (2024). In religious contexts, hallucinations can manifest as misquoted verses, fabricated Hadiths, or distorted interpretations, which pose significant ethical, theological, and social risks. Such errors may undermine public trust in AI systems and contribute to the spread of misinformation, particularly when dealing with sacred texts that have fixed, canonical forms.

Our main contributions to the IslamicEval-2025 Mubarak et al. (2025) shared task are threefold. First, we introduced a data pipeline to generate a synthetic dataset, enabling fine-tuning of a relatively small LLM (gpt-4.1-mini) for detecting spans of religious quotations—both claimed and correct. We benchmarked this approach against large reasoning models with access to a code interpreter, showing that the fine-tuned small model is cheaper and faster while maintaining strong performance. Second, we designed a layered hierarchical index and search algorithm, coupled with a low-cost LLM judge (gpt-4.1-mini), which outperformed a frontier reasoning model (GPT-5 with code interpreter) that is significantly slower and more expensive. Third, we developed a Retrieval-Augmented Generation (RAG) pipeline specialized for Quranic and Hadith question answering, tailored to the unique linguistic and semantic challenges of Islamic texts. We have released our GitHub repository publicly to facilitate transparency and reproducibility of our work [1].

2 Background

We participated in Subtask 1A, which takes a model response as input and detects spans labeled Ayah or Hadith. In addition, we participated in Subtask 1B, which validates the spans identified in Subtask 1A labeling it as correct or incorrect, while Subtask 1C

[1] https://github.com/sakher/IslamicEval-BurhanAI-Public

Proceedings of The Third Arabic Natural Language Processing Conference, pages 503–508
November 8-9, 2025 ©2025 Association for Computational Linguistics

corrects any spans marked as incorrect by providing their correct form or flagging them as incorrect. Finally, Subtask 2 focuses on retrieving the top 20 answer-bearing citations from the Quran and Sahih Al-Bukhari given an Arabic question.

Many previous works have addressed hallucination in large language models using different approaches. One line of research applies Retrieval-Augmented Generation (RAG) B'echard and Ayala (2024); Alan et al. (2024); Khalila et al. (2025). Other studies focus on instruction tuning and prompt engineering techniques Barkley and van der Merwe (2024); Hikal et al. (2025). Further research highlights verification and fact-checking strategies Sibaee et al. (2024). Additionally, some works emphasize fine-tuning with human feedback Cheng et al. (2025); Lin et al. (2025). Together, these methods enable LLMs to function as more effective tools for factual verification and reliable information use.

3 System Overview

3.1 Subtask 1A – Span Detection:

We used two approaches; we fine-tuned gpt-4.1-mini to output religious text spans. For fine-tuning we constructed a balanced training corpus (460 training examples and 83 validation examples) through multi-stage synthesis combining competition development data (70%) with synthetic examples (30%) generated using gpt-4.1 [2].

Separately, we leveraged a reasoning model with access to a code interpreter, testing both frontier and smaller OpenAI models (see detailed results in Table 1). The model was instructed to detect spans resembling Quran or Hadith. Since LLMs struggle with precise character counting Fu et al. (2024), we enabled the code interpreter tool: whenever the model needed to compute exact offsets, it could generate Python code, which was then executed in a secure sandbox, and the resulting values were fed back into the model. This ensured reliable start and end indices for each span. Outputs were further constrained using the OpenAI API's structured output feature with a JSON schema requiring a list of citations labeled as Ayah or Hadith with character offsets. We then applied heuristic post-processing: checking context within ±64 characters for lexical cues to refine labels, trimming extraneous punctuation or quotations, and merging or disentangling nested spans[3].

3.2 Subtask 1B – Validation and Subtask 1C – Correction:

Our system uses a layered design that combines seven forms of indexing with a six-stage search process. On the indexing side, every Quran verse and Hadith is indexed in multiple ways so the system can quickly switch between exact and approximate lookups. We keep exact MD5 hashes of the raw text, normalized versions without diacritics or punctuation, and character n-grams (3-grams by default) for fuzzy matches. Texts are also grouped into buckets by length to speed up candidate filtering, and we maintain a list for edit-distance checks. When available, we add a Whoosh full-text index for keyword search and a vector index built from Cohere embeddings stored in Qdrant for semantic similarity.

Searching happens in a strict sequence, with early stopping once a confident match is found. It starts with exact and normalized lookups, then falls back to n-gram fuzzy search. If needed, it escalates to semantic retrieval with embeddings and re-ranking. Next, it applies string-level fuzzy scorers such as Levenshtein distance and partial substring matching, followed by token-overlap checks to catch paraphrases. As a last resort, it computes Jaccard similarity on character trigrams. This stepwise design ensures clean matches are resolved instantly, while noisy, partial, or corrupted quotations are still recovered through progressively more flexible methods.

For the 1C correction subtask, we also tested a separate approach using a reasoning model - GPT-5 with high reasoning effort with access to tools. We give the model access to a code interpreter tool and to the corpora as text files. The model could perform multiple text-matching searches in the files to find the right match, then decide whether the matches were found to return them in JSON format.

3.3 Subtask 2

For Subtask 2, we built a Retrieval-Augmented Generation (RAG) system that retrieves passages from the Quran and Sahih Al-Bukhari. The corpora were split into 1,500-token chunks with 400-token overlap and stored as a vector dataset, allowing the reasoning model (GPT-5 with high reasoning) to run multiple searches per query when needed. The

[2] data generation pipeline https://github.com/sakher/IslamicEval-BurhanAI-Public/blob/main/abubakr/taskA/01-index-religion-dataset-for-search.py

[3] Prompts details https://github.com/sakher/IslamicEval-BurhanAI-Public/blob/main/task_a_prompt_engineering/pipeline_task_a.py

model could reformulate queries across iterations and returned ranked citations based on how directly and completely they answered the question. A deterministic post-processing pipeline then mapped Quran ayat to QPC Malhas and Elsayed (2020) passage IDs, validated hadith IDs against the official JSONL, removed duplicate citations.

4 Experimental Setup

All results were against test dataset and as seen on CodaBench. Our systems included a fine-tuned span detector (gpt-4.1-mini, 3 epochs, batch size 1, LR multiplier 2.0, temp 0). Implementation utilizes **Whoosh** for inverted indexing, **FuzzyWuzzy** for edit distance computation, **Qdrant** for vector storage, **Cohere embed-v4.0** for embeddings, **Cohere rerank-v3.5** for neural re-ranking, and **GPT-4.1-mini** for expert-guided validation for subtasks 1B and 1C. For evaluation, we used the proposed shared task evaluation metrics.

5 Results

Our system achieved a macro-averaged F1 score of 87.78 % using Fine-tuned a Span Detection Model approach, and 90.06% using reasoning model with access to tools (o4-mini model with high reasoning setting), see Table 1.

Although we tested larger models like the full-size o3 and GPT-5 three different sizes (full, mini and nano) with all reasoning levels (high, medium and low), none of these made it to the top 3 results, which shows that smaller models and fine-tuned tiny models can outperform larger models for such specialized tasks A.1.2.

Approach	Macro-Averaged F1
Approach-1	90.06%
Approach-2	87.78 %
Approach-3	87.10 %

Table 1: Task 1A evaluation results. Approach 1 is an OpenAI o4-mini with high reasoning effort reasoning model with access to tools. Approach 2 is an OpenAI o3-mini with high reasoning effort reasoning model with access to tools. Approach 3 is a fine-tuned gpt-4.1-mini span-detection model.

As for Subtask 1B, the layered hierarchical index and search algorithm achieves computational efficiency through exact matching optimization (constant-time hash operations) while maintaining comprehensive recall via semantic search for challenging disambiguation cases, yielding validation accuracy of 88.60% Table 2.

Approach	Accuracy
Hierarchical search-1	88.60 %

Table 2: Task 1B evaluation Accuracy results using layered hierarchical index and search algorithm with LLM-based validation.

For the Subtask 1C see Table 3, we used two approaches: layered hierarchical index and search algorithm with 66.56 % accuracy see section 3.2, and reasoning model with tools with 61.04 % accuracy. Table 3.

Approach	Accuracy
Hierarchical search-2	66.56 %
Reasoning model	61.04 %

Table 3: Subtask 1C evaluation Accuracy results. Hierarchical search-2 is a hierarchical search using a layered hierarchical indexing and search algorithm with LLM-based correction, and Reasoning model is a GPT-5 with high reasoning effort model with access to tools and post-processing.

For Subtask 2 see Table 4, the Mean Average Precision (MAP) was used as the main official measure for evaluation. We submitted only one submission. The results show that the model has some ability to find and rank relevant information, but there is significant room for improvement, especially for hypotheses.

Approach	MAP@10	MAP_Q@5	MAP_H@5
RAG-based(benchmark)	0.2807	0.3257	0.2386

Table 4: Subtask 2 results.

6 Conclusion

In this paper, we introduced an overview of our participation in the IslamicEval 2025 shared task Mubarak et al. (2025).

We proposed a layered hierarchical index and search algorithm with fine-tuned model to solve the Subtask 1A, 1B, 1C and reasoning model with tools for tasks 1A and 1C.

Our findings demonstrate that structured tool-assisted reasoning, hierarchical indexing with progressive search strategies, targeted fine-tuning of models, rigorous text normalization, corpus-restricted retrieval, and structured outputs are highly effective for mitigating hallucinations and ensuring precise retrieval in religious QA contexts. Crucially, our results highlight that compact fine-tuned models (such as GPT-4-mini) and, separately, smaller reasoning models (e.g., o4-mini) with tool access can each achieve comparable or superior performance to large, computationally expensive frontier systems (e.g., GPT-5 with high reasoning), significantly reducing cost and latency—particularly in specialized tasks like span detection and correction (Subtasks 1A and 1C)

In future work, we plan to:

1. Explore vector store ingestion strategies (chunk sizing, overlap) and Arabic-specialized embedding models to improve recall on paraphrastic questions.

2. Add optional query-expansion prompts (synonyms, tafsir-guided paraphrases) while retaining closed-world constraints.

3. Consider shallow re-ranking informed by lightweight heuristics (entity match, directive/answer verbs) only if it demonstrably preserves "answer-enclosing" priority.

4. Evaluate adding auxiliary corpora (e.g., tafsir) as side channels for query reformulation without polluting the scoring universe.

5. Expand the vector store with texts with and without tashkeel (diacritics).

Limitations

Due to the limited time of our submission, we conducted limited experiments to solve the shared task and we were not able to explore more solution spectrum. Consequently, we did not go in depth into the hallucination categories for more fine-grained solutions. The integration of RAG introduces dependencies on retrieval accuracy and system latency, which can constrain its applicability in real-time scenarios or in environments with limited or no connectivity. Although we utilized LLMs to detect hallucinations, we have not yet investigated hallucination occurrences within the generated solutions. Finally, using large frontier models with high reasoning requirements can be both computationally expensive and time-consuming. Therefore, our future work will focus on leveraging lightweight models to improve efficiency.

AI disclaimer

We used ChatGPT and Cursor under author supervision to assist with phrasing and to generate support code for boilerplate and utilities; all research ideas, algorithms, experimental design, and interpretations are the authors' own, and the authors reviewed all outputs and accept full responsibility for the code and text; **no AI system is an author**.

References

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydin. 2024. A rag-based question answering system proposal for understanding islam: Mufassirqas llm. *ArXiv*, abs/2401.15378.

Liam Barkley and Brink van der Merwe. 2024. Investigating the role of prompting and external tools in hallucination rates of large language models. *ArXiv*, abs/2410.19385.

Patrice B'echard and Orlando Marquez Ayala. 2024. Reducing hallucination in structured outputs via retrieval-augmented generation. In *North American Chapter of the Association for Computational Linguistics*.

Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, and Jiahui Wen. 2025. Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking. *ArXiv*, abs/2501.01306.

Tairan Fu, Raquel Ferrando, Javier Conde, Carlos Arriaga, and Pedro Reviriego. 2024. Why do large language models (llms) struggle to count letters? *arXiv preprint arXiv:2412.18626*.

Amina El Ganadi, Sania Aftar, Luca Gagliardelli, and Federico Ruozzi. 2025. Generative ai for islamic texts: The eman framework for mitigating gpt hallucinations. In *International Conference on Agents and Artificial Intelligence*.

Baraa Hikal, Ahmed Nasreldin, and Ali Hamdi. 2025. Msa at semeval-2025 task 3: High quality weak labeling and llm ensemble verification for multilingual hallucination detection. *ArXiv*, abs/2505.20880.

Ben Hutchinson. 2024. Modeling the sacred: Considerations when using considerations when using religious texts in natural language processing. In *NAACL-HLT*.

Zahra Khalila, Arbi Haza Nasution, Winda Monika, Aytuğ Onan, Yohei Murakami, Yasir Bin Ismail Radi, and Noor Mohammad Osmani. 2025. Investigating retrieval-augmented generation in quranic studies: A study of 13 open-source large language models. *ArXiv*, abs/2503.16581.

Johnny Li, Saksham Consul, Eda Zhou, James Wong, Naila Farooqui, Yuxin Ye, Nithyashree Manohar, Zhuxiaona Wei, Tian Wu, Ben Echols, Sharon Zhou, and Gregory Diamos. 2024. Banishing llm hallucinations requires rethinking generalization. *ArXiv*, abs/2406.17642.

Shuyuan Lin, Lei Duan, Philip Hughes, and Yuxuan Sheng. 2025. Harnessing rlhf for robust unanswerability recognition and trustworthy response generation in llms. *arXiv preprint arXiv:2507.16951*.

Rana Malhas and Tamer Elsayed. 2020. Ayatec: building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(6):1–21.

Marryam Yahya Mohammed, Sama Ayman Ali, Salma Khaled Ali, Ayad Abdul Majeed, and Ensaf Hussein Mohamed. 2025. Aftina: enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2024. Llms know more than they show: On the intrinsic representation of llm hallucinations. *ArXiv*, abs/2410.02707.

Serry Sibaee, Abdullah I. Alharbi, Samar Ahmed, Omar Nacar, Lahouri Ghouti, and Anis Koubaa. 2024. Asos at arabic llms hallucinations 2024: Can llms detect their hallucinations :). In *OSACT*.

A Appendix

A.1 Subtask 1A

A.1.1 Fine tune model

The Figure 1 presents the loss curve obtained from the fine-tuning process on the OpenAI platform. It shows the training loss progression for the fine-tuning configuration, illustrating a gradual convergence over the training steps. This plot provides insight into the stability and efficiency of the fine-tuning process.

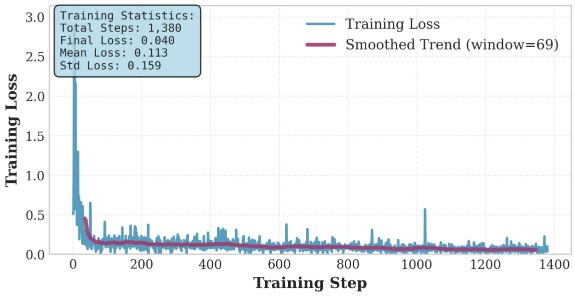


Figure 1: The plot generated from the fine-tuning loss table provided by the OpenAI platform.

A.1.2 Subtask 1A: Details results for reasoning model approach:

Model Name	Reasoning Effort	Score
GPT-5 Nano	low	0.82
O3	high	0.82
GPT-5	low	0.81
GPT-5 Nano	high	0.81
GPT-5 Nano	medium	0.81
O4 Mini	high	0.81
GPT-5	high	0.79
O3 Mini	high	0.79
GPT-5	medium	0.77
GPT-5 Mini	high	0.76
GPT-5	low	0.70
GPT-5 Mini	medium	0.65
GPT-5 Mini	high	0.63

Table 5: Performance of the AI reasoning model with access to tools was evaluated under varying levels of reasoning effort, using models of different sizes

From Table 5, we note that smaller models and tiny models can outperform larger models for such specialized tasks.

A.2 Subtask 2 evaluation on train and evaluation datasets

For Subtask 2 see Table 6, we use the organizers' code unmodified. Because train/dev lack hadith gold, our combined qrels capture Quran supervision only; hadith_sample.qrels remains empty, hence MAP_H@5 is 0 by construction. Evaluation results (merged train + dev Qrels): MAP@10=0.6199, MAP_Q@5=0.5761, MAP_H@5=0.0000 (expected given missing hadith gold).

Approach	**MAP@10**	**MAP_Q@5**	**MAP_H@5**
RAG-based(dev+train datasets)	0.6199	0.5761	0.0000
RAG-based(benchmark)	0.2807	0.3257	0.2386

Table 6: Subtask 2 evaluation results.

HUMAIN at IslamicEval 2025 Shared Task 1: A Three-Stage LLM-Based Pipeline for Detecting and Correcting Hallucinations in Quran and Hadith

Arwa Omayrah **Sakhar Alkhereyf** **Ahmed Abdelali**

Abdulmohsen Althubaity **Jeril Kuriakose** **Ibrahim AbdulMajeed**
HUMAIN, Saudi Arabia
aomayrah,salkhereyf@humain.com

Abstract

This paper presents HUMAIN's submission to the IslamicEval 2025 Shared Task 1, addressing hallucination detection and correction in Quranic and Hadith LLM-generated content. Our three-stage pipeline covers: (1) Span Detection via sequence-to-sequence annotation using TANL-style markup, (2) Validation with retrieval-based similarity and substring matching against reference corpora, and (3) Correction through exact matching, LCS alignment, and semantic re-ranking. On the official test set, our system achieved 87.2% F-1 for span detection, 86.1% accuracy for validation, and 68.2% accuracy for correction. While systematic detection is highly achievable, meaningful correction remains limited by semantic complexity where small textual differences can significantly impact religious understanding. This work presents a multi-stage LLM-based pipeline for Islamic content verification.

1 Introduction

Large Language Models (LLMs) enable advanced text generation but suffer from hallucination—producing linguistically fluent yet factually incorrect text. While problematic across domains, hallucinations pose critical risks in religious contexts, especially for the Quran and Hadith, where accuracy is essential. Even small errors (e.g., incorrect verse numbering, misattribution) may propagate misleading teachings or erode trust.

The **IslamicEval 2025 Shared Task** (Mubarak et al., 2025) addresses this by benchmarking hallucination detection and correction for Quranic and Hadith content. HUMAIN participated in Subtask 1 (A: Span Detection, B: Span Validation, and C: Span Correction). We propose a **three-stage** pipeline integrating sequence annotation, retrieval-based verification, and correction via semantic re-ranking. Our system achieved competitive results across all subtasks, highlighting both strengths and limitations of current LLM approaches. We made our system codes public on GitHub [1]. We have included our codes, prompts, and implementation details in our GitHub repository.

The paper is structured as follows: section 2 outlines the shared task setup. Section 3 describes our system architecture. Section 4 details experimental settings. Section 5 reports results, and section 6 concludes the paper with insights and future directions.

2 Background

The IslamicEval 2025 Shared Task (Mubarak et al., 2025) was designed to to evaluate system performance on hallucination detection and correction of Quranic and Hadith content produced by LLMs. The focus is on ensuring factual accuracy in religious texts, where even minor deviations are unacceptable.

2.1 Task Setup

Our team participated exclusively in Subtask 1, covering all three subtasks:

- **1A – Span Detection:** Identify spans in LLM outputs that correspond to Quranic verses or Hadith. This requires handling varied quotation styles, partial matches, and noise from generative models.
- **1B – Validation:** Determine whether each detected span is authentic and correctly quoted by comparing against reference corpora (Quran and six Hadith Books).
- **1C - Correction:** For spans deemed incorrect, provide the corrected version from the gold-standard texts, or indicate that the span is completely wrong.

[1] https://github.com/0xArwa/humain-islamiceval-2025

Proceedings of The Third Arabic Natural Language Processing Conference, pages 509–514
November 8-9, 2025 ©2025 Association for Computational Linguistics

All datasets are in Arabic and sourced from authentic Quran and Hadith corpora curated by the organizers. Each subtask contains 50 and 104 distinct samples in the dev and test sets, respectively. Predictions were submitted through CodaBench for official scoring on the test set.

3 System Overview

3.1 Subtask 1A – Span Detection

For span detection, we employ an LLM-based pipeline to identify and extract Quranic verses and Hadith passages. More details on the LLMs used in our experiments are shown in section 4. The process begins with preprocessing the input text to resolve formatting inconsistencies—such as irregular spacing, punctuation issues, or line breaks—ensuring that the text is normalized before being passed to the model.

The cleaned text is then provided to an LLM with a specialized system prompt and few-shot examples. These instruct the model to detect religious spans and annotate them using a bracket-based notation of the form `[span_text|tag_type]`, where `span_text` represents the identified religious content and `tag_type` specifies whether it is a Quranic verse (ق) or a Hadith (ح). For example:

Input:

وجاء في الحديث الشريف: إِنَّمَا الأَعْمَالُ بِالنِّيَّاتِ

Output:

وجاء في الحديث الشريف: [إِنَّمَا الأَعْمَالُ بِالنِّيَّاتِ | ح]

Particularly, span detection is modeled as a sequence-to-sequence translation task using the Translation between Augmented Natural Languages (TANL) framework (Paolini et al., 2021). The model regenerates the passage with special markers denoting the start, end, and type of each span. Because generative models may introduce slight variations in spacing or punctuation (or removing/adding words), the TANL framework first cleans the annotated output by removing special tokens and discarding invalid formats. After this normalization, TANL employs the Needleman–Wunsch Dynamic Programming (DP) algorithm (Needleman and Wunsch, 1970) to align the cleaned output with the original input at the token level. This alignment enables each detected span to be mapped back to its precise character positions in the source text, ensuring consistency despite formatting drift introduced during generation.

As an alternative to TANL's alignment process, we also experimented with a guided decoding setup. In this variant, the LLM directly generates structured JSON output following a predefined schema, where each span object includes its type, textual content, and character indices. We utilize the vLLM library (Kwon et al., 2023) to enable guided decoding, which we apply only when we have direct access to the model and can deploy it on vLLM. This approach removes the need for token-level alignment altogether, since positional information is produced natively during generation.

3.2 Subtask 1B – Validation of Content Accuracy

For span validation, we developed a sophisticated verification system that handles both Quranic verses and Hadith texts through specialized processing pipelines optimized for each content type.

Hierarchical Indexing: The system employs dual indexing of reference corpora with normalized full-text indices for exact lookups and word-based inverted indices for candidate retrieval.

Verification Strategies: The core verification process implements multiple complementary matching approaches:

Multi-text Detection: The system first determines whether spans contain single or multiple verses using smart pattern detection that analyzes separators including asterisks (*), parenthetical verse numbers (e.g., (٤١)), sequences of 3+ consecutive non-Arabic characters, and contextual comma usage. This detection guides the subsequent verification approach.

Exact Matching: First-stage verification performs direct hash-based lookup in the normalized index for perfect matches after diacritic removal and character standardization.

Strict Substring Matching: For cases requiring exact textual containment, the system verifies that the normalized input appears as a complete substring within reference texts. This approach proved particularly effective for Hadith validation where authentic partial quotations are common.

Fuzzy Matching: When exact methods fail, the system applies sequence matching algorithms with experimentally-determined longest common subsequence (LCS) (Hirschberg, 1975) similarity thresholds. The process includes candidate pre-filtering using word overlap to reduce computational complexity, followed by detailed similarity scoring using LCS ratios.

Multi-word Substring Logic: For spans containing multiple words, specialized logic determines whether the entire sequence appears as a coherent substring in longer reference texts, with enhanced similarity scoring for valid substring matches.

Content-Specific Optimization: Based on empirical evaluation, we configured different verification approaches for each content type. Quranic spans use fuzzy matching with LCS similarity thresholds above 0.85 to maintain strict accuracy requirements for sacred text. Hadith spans employ strict substring matching, which better accommodates the legitimate partial quotations and paraphrasing patterns found in authentic Hadith transmission.

For multi-text spans, individual components are verified separately and aggregated using configurable consensus strategies.

3.3 Subtask 1C – Error Correction

Span correction for potentially corrupted or incomplete Quranic and Hadith texts is addressed through a multi-stage pipeline. The process begins with index-based pre-filtering, which combines a word-level inverted index with a character 3-gram index to reduce the search space. This design captures both exact word matches and partial substrings, ensuring that noisy or fragmented queries still retrieve relevant candidates.

Immediately after pre-filtering, the pipeline applies a composite fallback scoring mechanism to handle edge cases such as queries that span multiple consecutive verses presented as continuous strings without separators, or minor lexical variations that prevent standard matches. This mechanism integrates word n-gram overlap, phrase continuity, and substring containment metrics, adjusting candidate scores to ensure that these cases are retained and prioritized in subsequent processing.

Following this early edge-case handling, the candidate spans proceed to three successive matching stages. The first stage performs exact substring matching on normalized text, returning immediate matches when the query sequence appears exactly after diacritic and punctuation removal. The second stage applies LCS algorithm with source-specific similarity thresholds (Quran $\geq$ 0.85, Hadith $\geq$ 0.75). The third stage employs a multilingual semantic reranker (bge-reranker-v2-m3) (Chen et al., 2023) that applies sigmoid activation to produce normalized semantic similarity scores between 0 and 1 for the top candidates from earlier stages. The reranker evaluates semantic similarity beyond lexical overlap, combining its scores with original LCS similarities using a weighted scheme ($\alpha = 0.7$). This hybrid approach promotes semantically correct matches that may have lower lexical overlap, addressing cases where authentic content differs significantly in wording from the query.

4 Experimental Setup

4.1 Data Preprocessing

All input texts were normalized including diacritic elimination, character variant normalization (e.g. أ، إ، آ → ا), punctuation elimination, and whitespace standardization to ensure consistent matching across various text formats.

4.2 Model Configurations

For **Subtask 1A**, we experimented with various LLMs: GPT-4o (via OpenAI API) and four Arabic-centric LLMs, ALLAM (Bari et al., 2024), Fanar (Team et al., 2025), Command-R7B (Alnumay et al., 2025), and Jais-13B (Sengupta et al., 2023), all without task-specific fine-tuning (*temp*=0.1, *top_p*=0.98). For **Subtask 1B**, the selected similarity thresholds are $\geq$ 0.9 for Quran and strict substring matching for Hadith. For **Subtask 1C**, we combine the reranker (top-20) with final similarity thresholds set to $\geq$ 0.85 for Quran and $\geq$ 0.75 for Hadith, with spans below marked as خطأ (uncorrectable).

5 Results

This section shows the results of our system on the three subtasks of IslamicEval 2025.

5.1 Subtask 1A: Span Detection

Model	Dev			Test
	P	R	F1	F1
GPT-4o	87.4	75.7	81.1	**87.2**
ALLAM-34B	79.5	75.0	77.2	78.1
Command-R7B-Arabic	62.6	39.1	48.1	-
Fanar(API)	32.0	23.3	27.0	-
Jais-13b-chat	16.8	10.5	12.9	-

Table 1: Subtask 1A: Span Detection Performance. (P: Precision, R: Recall).

Table 1 shows the character-level macro-averaged F-1 scores for the five LLMs on the dev set. From these, we selected only the top-performing two models for submission on CodaBench (i.e., for the test set).

Quran Performance				
Configuration	**Acc.**	**P**	**R**	**F1**
Strict	88	95	85	90
Fuzzy(0.9)	**91**	93	**93**	**93**
Fuzzy(0.65)	88	86	**97**	91

Hadith Performance				
Configuration	**Acc.**	**P**	**R**	**F1**
Strict	**85**	97	**76**	**85**
Fuzzy(0.8)	75	**100**	54	70
Fuzzy(0.25)	75	**100**	54	70

Table 2: Substring matching performance comparison for Quran and Hadith text verification. Parentheses indicate LCS similarity thresholds. Fuzzy matching works better for Quran due to textual variations (different Uthmani formats, with/without tashkeel diacritics), while strict matching is optimal for Hadith due to text standardization. The distinct optimal strategies reflect the nature of each corpus: Quran exists in multiple valid variants requiring flexible matching, whereas Hadith collections maintain consistent formatting.

On the test set, GPT-4o achieved 87.20% F-1, while ALLAM-34B reached 78.10%, demonstrating competitive performance under more constrained settings. Both GPT-4o and Fanar (API) employed the prompting with special markers as described earlier in subsection 3.1 as we did not have access to them. For other 3 models, we utilized the guided decoding approach to ensure structured JSON output generation. Among these, Command-R7B-Arabic was the second-best performing Arabic-centric model, though it lagged significantly behind ALLAM in overall accuracy. Jais and Fanar showed considerably lower performance, indicating that current smaller Arabic-centric models are not yet competitive for this task.

Importantly, all results were obtained without any fine-tuning of model weights, showing that our approach can generalize to different LLMs without expensive adaptation.

5.2 Subtask 1B: Validation of Content Accuracy

As described in subsection 3.2, our system supports both fuzzy substring matching (with configurable LCS similarity thresholds) and strict substring matching. Table 2 presents development set performance across different threshold configurations by content type, which guided our optimal configuration selection for test evaluation.

Table 3 shows performance of selected configurations, where "–" indicates strict substring matching (no threshold required). The repeated Hadith values demonstrate substring matching robustness across threshold combinations. Our optimal configuration achieved 86.14% test accuracy using fuzzy substring matching with a 0.90 LCS similarity threshold for Quranic content and strict substring matching for Hadith texts.

This hybrid approach addresses different validation requirements: Quranic verses need high LCS similarity thresholds for fuzzy matching to handle script variations between Uthmani and formal scripts while maintaining accuracy against the Uthmani reference corpus, whereas Hadith texts benefit from exact substring matching for partial quotations and paraphrasing.

Parameters		Performance	
Quran	**Hadith**	**Dev (%)**	**Test (%)**
0.80	0.65	84.21	84.21
0.90	0.80	81.00	85.96
0.90	–	84.60	**86.14**

Table 3: Subtask 1B: Span Validation Overall Performance Comparison.

5.3 Subtask 1C: Error Correction

The best configuration, which combines exact matching, LCS, and semantic reranking, achieved 68.18% test accuracy, substantially improving over simpler baseline as shown in Table 4. Overall, the system shows strong performance in span detection and validation, while error correction remains the most challenging aspect, suggesting the need for more semantically grounded approaches.

Method	Thresholds		Accuracy	
	Quran	**Hadith**	**Dev**	**Test**
LCS	0.70	0.65	54.60	57.48
EM+LCS	0.85	0.70	60.13	65.91
+Reranker	0.85	0.75	72.74	**68.18**

Table 4: Subtask 1C: Error Correction Performance with Different Methods (LCS: Longest Common Subsequence, EM: Exact Match).

6 Conclusion

This paper presents a comprehensive three-stage pipeline for detecting and correcting hallucinations in Quranic and Hadith content generated by LLMs, addressing a critical challenge where factual accuracy carries profound religious and cultural significance. Through our participation in the IslamicEval 2025 shared task, we demonstrate that specialized approaches can effectively handle the unique requirements of Islamic textual verification.

The results highlight several key insights. GPT-4o outperformed other models overall in span detection, while ALLAM showed the strongest performance among Arabic-centric models, indicating the growing maturity of regional LLMs. Importantly, our system achieves strong results without any fine-tuning, showing that the approach can be applied to different models without modifying their weights —an advantage in terms of cost and scalability. Different similarity thresholds are needed for Quran versus Hadith validation, and semantic reranking provides modest but consistent improvements over exact matching and LCS in correction tasks.

The relatively modest correction accuracy underscores the complexity of this task and the need for continued research. Our analysis reveals that the most challenging correction cases involve contextual misattributions where the hallucinated span shares thematic content with the correct reference but differs substantially in wording. For instance, spans discussing the same Quranic narrative may require corrections that are semantically related but lexically distant. In addition, some fabricated content is so disconnected from authentic sources that determining whether any meaningful correction exists presents significant challenges for automated systems, particularly when minimal lexical overlap (e.g., sharing only one or two common words) may result in questionable matches, where providing no correction might be more appropriate (subsection 1.2). This limitation highlights the need for context-aware correction methods that consider not just the isolated span but also its surrounding discourse and thematic coherence. Future work should focus on proactive hallucination prevention, integration of Islamic scholarly expertise, and development of more sophisticated retrieval-augmented generation systems. This research represents a crucial step toward building trustworthy AI systems for religious texts, where accuracy is not merely a technical requirement but a matter of profound cultural and spiritual significance. Our publicly available code contributes to ongoing efforts in this critical domain.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback, the shared task organizers for their efforts, and HUMAIN for providing access to various models and infrastructure.

References

Yazeed Alnumay, Alexandre Barbet, Anna Bialas, William Darling, Shaan Desai, Joan Devassy, Kyle Duffy, Stephanie Howe, Olivia Lasche, Justin Lee, and 1 others. 2025. Command R7B Arabic: A small, enterprise focused, multilingual, and culturally aware Arabic llm. *arXiv preprint arXiv:2503.14603*.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan AlRashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, and 6 others. 2024. ALLaM: Large language models for Arabic and English. *Preprint*, arXiv:2407.15390.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2309.07597.

Daniel S. Hirschberg. 1975. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *Preprint*, arXiv:2101.05779.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, and 1 others. 2023. Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Fanar Team, 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, and 23 others. 2025. Fanar: An Arabic-centric multimodal generative ai platform. *Preprint*, arXiv:2501.13944.

A Appendix

1.1 Guided Decoding: Schema Definition

As an alternative to TANL's post-processing alignment, we experimented with guided decoding to constrain the model to produce valid JSON conforming to our span detection schema.

The model generates spans with explicit character positions, formally described by the following JSON Schema:

```
{
  "type": "object",
  "properties": {
    "spans": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "type": {
            "enum": ["q", "h"]
          },
          "text": {
            "type": "string"
          },
          "start": {
            "type": "integer"
          },
          "end": {
            "type": "integer"
          }
        },
        "required": [
          "type", "text",
          "start", "end"
        ]
      }
    }
  }
}
```

where type denotes Quran (q) or Hadith (h), and start/end specify character indices.

1.1.1 Example

For input text إنا أنزلناه في ليلة القدر والله أعلم,
the model generates:

```
{
  "spans": [
    {
      "type": "q",
      "text": "إنا أنزلناه في ليلة القدر",
      "start": 0,
      "end": 25
    }
  ]
}
```

1.1.2 Limitation

This approach relies entirely on the model's ability to generate accurate character positions during inference. The guided decoding constraints (via vLLM's guided_json parameter) ensure structural validity but cannot prevent hallucination of non-existent text spans.

1.2 Challenging Correction

This section illustrates cases where automated correction systems face significant challenges in determining appropriate mappings between fabricated content and authentic sources.

Example: Hallucinated verse with minimal lexical overlap

LLM-Generated (Hallucinated):

قوله تعالى: ۡوإذا مرۡضت فلا ركب علي ولا
حمية وأصبح بياضۡ

Annotation Label:

WrongAyah (correctable)

Correction:

وإذا مرضت فهو يشفين

Analysis: The fabricated content shares minimal lexical overlap with the proposed correction (primarily the words إذا مرضت). The hallucinated verse contains nonsensical elements and bears no meaningful semantic relationship to the authentic verse. This example demonstrates the challenge of determining when shared vocabulary constitutes sufficient grounds for correction versus when providing no correction (خطأ) might be more appropriate.

TCE at IslamicEval 2025: Retrieval-Augmented LLMs for Quranic and Hadith Content Identification and Verification

Mohammed ElKoumy[1] **Khalid Allam**[2] **Ahmed Tamer**[2] **Mohammed Elqabalawy**[2]
[1]Rensselaer Polytechnic Institute, Department of Computer Science, Troy, NY, USA
[2]Tanta University, Computer and Control Department, Tanta, Egypt
elkoum@rpi.edu khalidallam222@gmail.com ahmad.tamer1908@gmail.com
mohammed30971488@f-eng.tanta.edu.eg

Abstract

Recent advancements in large language models (LLMs) have opened new possibilities for processing complex natural language tasks, including those involving highly regarded religious content. However, working with divine sources such as the Holy Quran and Hadith presents unique challenges. These Classical Arabic texts have, for centuries, been meticulously preserved and recited word-for-word, allowing no tolerance for errors — even a single incorrect diacritic can entirely alter the meaning. Such sensitivity demands exceptional precision, as hallucinations or inaccuracies from LLMs could lead to significant misinterpretations among general users. To address this challenge, we present an Arabic-focused, LLM-powered framework designed to identify and verify the integrity of religious text generated by widely used LLMs. Evaluation on benchmark subtasks demonstrates strong performance, achieving a Macro-Avg F1 score of **86.11%** on **Subtask 1A** and an Accuracy of **89.82%** on **Subtask 1B**.

1 Introduction

With the superior text generation capabilities of contemporary (LLMs) (Ouyang et al., 2022; OpenAI team, 2024), inaccurate yet plausible content, commonly known as hallucinations, has proliferated across various online platforms and websites (Huang et al., 2025). In response, the research community has developed fact-checking and verification methods grounded in reliable factual resources (Guo et al., 2022; Althabiti et al., 2024).

Given that languages reflect cultures, some of the content generated by LLMs in the Middle East is closely tied to the region's rich Islamic heritage, especially as these models are increasingly used for everyday tasks (Bashir et al., 2023; Mubarak et al., 2025). Consequently, there is a risk that fabricated sacred Islamic content may be generated and mistakenly treated as authentic or employed to reinforce Islamophobia or misinformation. This problem is particularly sensitive due to its significance among Muslim and Arab communities (Mubarak et al., 2025).

In this paper, we present our approach to address these challenges by focusing on the tasks of Islamic content identification and validation, namely **Subtask 1A** and **Subtask 1B** of IslamicEval (Mubarak et al., 2025), respectively.

Given the limited size of the dataset and minimal financial and time resources, our approach adopts a few-shot learning strategy powered by state-of-the-art (SOTA) LLMs to address both tasks (Liu et al., 2023; Ouyang et al., 2022). More specifically, to automatically identify divine texts in **Subtask 1A**, we leverage trigger words and common citation patterns frequently found in religious content (Bashir et al., 2023). For the verification subtask, i.e, **Subtask 1B**, we employ a retrieval-augmented LLM architecture with integrated content validation, enabling precise cross-checking of generated text against authoritative Islamic sources (Guo et al., 2022; Mubarak et al., 2025).

Our contributions are as follows:

- We achieve strong performance on both tasks using powerful multilingual LLMs such as Qwen-235B (MoE) and GPT-4o. Our results confirm that a carefully designed prompt can lead to superior performance across both tasks.

- To make verification feasible despite the large size of the authentic reference resources, we employ a lexical matching system to retrieve the most relevant verses and implement an efficient early exit strategy once verification is successful. In addition, we empirically demonstrate the effectiveness of this retrieval phase.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 515–527
November 8-9, 2025 ©2025 Association for Computational Linguistics

- We validate the consistency of our results by demonstrating strong agreement between the development set performance and the hidden final test dataset.
- We share our code[1] with the community to promote broader accessibility and encourage further exploration and improvement on this essential problem.

We organize the paper as follows. Section 2 presents the background for the task and related literature. In Section 3, we provide a detailed description of the system design for both tasks. Subsequently, Section 4 highlights the key experimental details and running configurations. Section 5 presents the results along with analysis and findings. Finally, Section 6 concludes the work.

2 Background

Attention mechanisms and the Transformer architecture have revolutionized NLP by enabling models to effectively capture long-range dependencies (Vaswani et al., 2017). Models like BERT and its Arabic variants, e.g., AraBERT (Antoun et al., 2020), have further showcased the success of these advancements for both multilingual and Arabic NLP tasks (Devlin et al., 2019). Recently, LLMs such as GPT-4 have demonstrated impressive few-shot learning capabilities (Brown et al., 2020; OpenAI team, 2024), allowing them to perform a wide range of tasks with minimal task-specific tuning. Meanwhile, prompt engineering has emerged as a crucial technique to tailor these powerful models to specialized applications (Liu et al., 2023).

Accurate processing of Islamic sacred texts is essential due to their cultural and religious significance in the Arabic world. NLP tasks targeting these texts include question answering (QA), content retrieval, morphological analysis, and recitation correction, among others (Bashir et al., 2023). Prior shared tasks, notably Qur'an QA 2022 and 2023, have laid the groundwork by focusing on QA over the Noble Quran using retrieval and comprehension techniques (Malhas et al., 2022, 2023). Central to these efforts, retrieval methods based on lexical approaches such as TF-IDF and BM25 continue to play a fundamental role in effectively locating relevant verses or narrations (Salton and Buckley, 1988).

Building upon previous endeavors, IslamicEval 2025 tackles the critical challenge of hallucination detection in LLM-generated Islamic content, emphasizing the accuracy and integrity of Quranic and Hadith references (Mubarak et al., 2025). The competition comprises the following subtasks:

- **Subtask 1A: Identification** — Detect spans of Quranic verses (Ayahs) and Hadiths within free-text responses generated by LLMs.
- **Subtask 1B: Validation** — Assess each identified utterance against authoritative sources to distinguish accurate references from hallucinated content.
- **Subtask 1C: Correction** — Generate corrected versions of any erroneously generated Ayahs or Hadiths based on authentic sources.
- **Subtask 2: Passage Retrieval** — Retrieve a ranked list of Quranic or Hadith passages that potentially answer a given question posed in Modern Standard Arabic.

As previously noted, this work presents our solutions for the **1A** and **1B** subtasks. Detailed dataset statistics for both subtasks are provided in Tables 4 and 5 in Appendices A.1 and B.1 respectively.

3 System Design

Our approach leverages few-shot learning with SOTA foundational LLMs to address both subtasks. For **1B** subtask, we propose a retrieval-augmented architecture to perform the verification procedure.

3.1 Subtask 1A: Span Extraction For Identification

For this subtask, we formulate the problem as a span extraction task, where the system identifies textual segments referencing Quranic verses and Hadith within generated responses (Mubarak et al., 2025). Our approach employs a powerful foundational LLM (Yang et al., 2025), guided by a carefully designed few-shot prompt to extract relevant spans (OpenAI team, 2024; Liu et al., 2023; Brown et al., 2020). These prompts emphasize commonly occurring trigger words and citation patterns characteristic of sacred Islamic texts, enabling effective identification given the structured nature of the citation process (Bashir et al., 2023) (See Appendix A.2 for detailed prompts in Figure 3). To ensure the input remains manageable for the LLM

[1] The code and resources are available at https://github.com/m-alqblawi/Islamic_Eval_2025

while preserving essential information, we apply chunking to segment the input into appropriately sized portions.

Driven by the limited size of the training dataset, we forego extensive task-specific fine-tuning and instead leverage the strong generalization capabilities of foundational LLMs for span extraction (Devlin et al., 2019; Vaswani et al., 2017). Subsequently, to accurately align the extracted spans with their precise locations in the generated text, the system incorporates a fuzzy matching module (Platenius et al., 2013; Salton and Buckley, 1988) that accounts for minor variations and inconsistencies. Spans with fuzzy matching scores below a predefined threshold are discarded to maintain high precision and minimize false positives. The complete system architecture is illustrated in Figure 2, with algorithmic details provided in Algorithm 1 in Appendix A.2.

3.2 Subtask 1B: Retrieval-Augmented Verification

Our approach for Subtask **1B** consists of three main phases that integrate a powerful foundational LLM with a retrieval mechanism tailored for Quranic and Hadith verification. We model this subtask as two independent few-shot binary classification problems — one for Quran verification and another for Hadith verification.

First, we retrieve relevant passages from authenticated Quranic and Hadith sources using a hybrid retrieval strategy. For Quranic material, retrieval leverages fuzzy matching based on the `py_quran` Python package (Yousef et al., 2018). Our approach performs verse-level retrieval by tokenizing the query into individual words and computing a weighted matching score for each verse based on the frequency and presence of these words. Specifically, a voting or counting map is constructed where each word match contributes to the verse's overall relevance score, allowing the system to identify the most pertinent verses despite minor textual and scripting variations.

For Hadith content, we employ a character-level TF-IDF ranking approach with character n-grams to capture fine-grained textual patterns (Salton and Buckley, 1988). After retrieval, a postprocessing algorithm is applied to the Quranic results to merge adjacent retrieved verses from the same surah into coherent contiguous segments, enhancing context and verification accuracy before input to the LLM. Subsequently, these consolidated retrieval results form the input context for the LLM, which determines the correctness of the claims through few-shot prompting. In our prompt template, we provide few-shot demonstration examples independently for Quranic and Hadith texts (detailed prompts shown in Figure 5 in Appendix B.2).

For Quran verification, the LLM is tasked with strict word-for-word matching due to the sensitivity of small textual changes on meaning (Bashir et al., 2023). In contrast, Hadith verification tolerates minor variations in the *matn* (narrative text), acknowledging authentic variations in Prophetic sayings.

It is worth mentioning that the verification LLM is invoked sequentially on each retrieved result independently. If a match is found by the LLM, the sequential process terminates early, mirroring the human strategy of stopping once sufficient evidence is found. A comprehensive system architecture is presented in Figure 4, with the detailed algorithmic implementation described in Algorithm 2 in Appendix B.2.

4 Experimental Details

For both tasks, we utilized the original development and test splits. Due to constraints in budget and time, our experiments did not involve exhaustive exploration of all possible parameters and configurations. We leave this comprehensive investigation to future work and the community.

For Subtask **1A**, we employ various multilingual Qwen and LLaMA3 LLMs (Yang et al., 2025; Grattafiori et al., 2024), accessing all open-source models via the Hugging Face API. The LLM is prompted to output the extracted spans in a structured JSON format. The fuzzy matching threshold is set to a high value of 90% for precise matching and robustness against hallucination. Chunking was applied consistently throughout all experiments using sentence-aware segmentation with a 800-character limit, preserving semantic boundaries at Arabic and standard punctuation marks. To assess the impact of this technique, we conducted a minor ablation study by disabling chunking for our top-performing model.

In Subtask **1B**, for Quran retrieval, since citations must be exact word-by-word matches, we combine a proximity score between matched words to preserve their relative order, along with a coverage score representing the proportion of matched words within each potential ayah. For Hadith retrieval, we employ a TF-IDF module configured with character n-grams up to 7-grams to capture fine-grained textual patterns. For verification, we experimented with two distinct models; we utilized the open-source Gemma model (Team et al., 2024), accessed through OpenRouter, alongside GPT-4 via the OpenAI API (OpenAI team, 2024).

5 Results and Analysis

Subtask 1A: Table 1 presents the validation set performance for span extraction across different LLMs. The Qwen3-235B-A22B-Instruct (MOE) model achieves the best overall performance with an accuracy of **0.860** and a macro-average F1 score of **0.765**, demonstrating superior capability in identifying Islamic content spans. Notably, the comparison between the chunked and non-chunked versions of the same model reveals the significant impact of preprocessing: the model without chunking achieves substantially lower performance (0.795 accuracy vs. 0.860), confirming the importance of chunking preprocessing for maintaining model performance on longer text inputs. Among smaller models, Qwen14B shows competitive precision (0.807), while Llama-3.3-70B-Instruct lags behind other models across all metrics.

Subtask 1B: Table 2 shows the binary classification results for Islamic content verification. GPT-4o with Arabic diacritics achieves the highest performance with an accuracy of **0.9** and F1 score of **0.92**, significantly outperforming all Gemma variants. Among the Gemma models, the 12B variants consistently outperform 4B variants, with Gemma-12B-IT (with diacritics) achieving 0.737 accuracy compared to 0.676 for Gemma-4B-IT.

Our deeper analysis of these results reveals several critical insights: **(1)** The high recall rates achieved by the full pipeline across all experimental conditions (consistently above 95% as shown in Table 2) indicate that our hybrid retrieval architecture effectively captures relevant Islamic content from authoritative sources. However, as evidenced in Tables 6, 7, and 8, **(2)** we observe a consistent pattern toward Type I errors (false positives are underlined and italicized in all confusion matrices for clarity), suggesting that LLM verifiers are occasionally deceived by similar Islamic content generated by powerful language models.

(3) Removing diacritics generally reduces performance across all model sizes, with accuracy drops of 2-3 percentage points (e.g., Gemma-12B drops from 0.737 to 0.709). This performance degradation is particularly pronounced in Quranic content compared to Hadith content, especially for GPT-4o, suggesting that diacritical marks are essential for understanding nuanced Quranic text where subtle diacritical differences significantly impact meaning. **(4)** Verification errors are significantly more prevalent in Quranic content than in Hadith content, indicating that Quranic language presents greater verification challenges. This disparity stems from two key factors: first, the strict word-for-word preservation requirements in Quranic text compared to the relatively acceptable variations in Hadith transmission; and second, the precise linguistic requirements and rich diacritical structure inherent to Quranic Arabic. In contrast, Hadith content allows for authentic variations in transmission across different narrations, making it inherently more tolerant of minor textual discrepancies. Given GPT-4o's superior discriminative capabilities compared to open-source Gemma variants, these structural differences between Quranic and Hadith content explain why GPT-4o consistently produced the fewest errors across all verification tasks.

Official Test Set Performance: Table 3 reports the final results on the hidden test set as provided by the IslamicEval 2025 organizers. Our best-performing models, **Qwen3-235B-A22B-Instruct for Subtask 1A** and **GPT-4o for Subtask 1B**, achieved strong performance on the official evaluation: **0.861 macro-average F1** for span identification and **0.898 accuracy** for verification, respectively. These results demonstrate the effectiveness of our hybrid approach combining large language models with domain-specific preprocessing and retrieval strategies for Islamic content processing tasks.

Task	Metric	Score
1A (Qwen3-235B)	Macro F1	0.861
1B (GPT-4o)	Accuracy	0.898

Table 3: Official Test Results from IslamicEval 2025

6 Conclusion

We present a framework for identifying and verifying Islamic content in LLM-generated text, addressing hallucination detection in sacred Arabic texts. Our approach combines SOTA multilingual LLMs with domain-specific preprocessing and retrieval-augmented verification strategies.

Index	Model	Accuracy	Precision	Recall	Macro-F1
1	Qwen3-8B	0.836	0.778	0.766	0.751
2	Qwen14B	0.835	**0.807**	0.781	0.765
3	Qwen3-32B	0.804	0.795	0.772	0.758
4	Llama-3.3-70B-Instruct	0.731	0.743	0.698	0.700
5	Qwen3-235B-A22B-Instruct (MOE)	**0.860**	0.801	**0.789**	**0.765**
6	Qwen3-235B-A22B-Instruct (MOE)[†]	0.795	0.769	0.748	0.719

[†]Without chunking preprocessing step.

Table 1: Validation Set Performance for Official Split on Subtask 1A. Models are ordered by parameter size from the smallest to largest.

Index	Model	Accuracy	Precision	Recall	F1
1	Gemma-4B[†]	0.664	0.642	**0.986**	0.777
2	Gemma-4B	0.676	0.652	0.980	0.783
3	Gemma-12B[†]	0.709	0.674	0.986	0.801
4	Gemma-12B	0.737	0.697	0.986	0.817
5	GPT-4o[†]	0.87	0.82	**0.986**	0.9
6	**GPT-4o**	**0.9**	**0.87**	**0.986**	**0.92**

[†]Without diacritics.

Table 2: Validation Set Performance for Subtask 1B

Our results demonstrate strong performance: **86.11%** macro-average F1 on Subtask **1A** and **89.82%** accuracy on Subtask **1B**. Key findings include the critical importance of chunking preprocessing for longer text inputs. The retrieval-augmented approach enables precise cross-checking against authoritative sources while maintaining computational efficiency through early termination strategies.

This work contributes to the broader effort of ensuring accuracy and integrity in AI-generated religious content, addressing a critical need for the Muslim community. We hope our publicly available code and findings facilitate further exploration and improvement in this essential domain.

Limitations

As noted in prior studies (Farghaly and Shaalan, 2009; Bashir et al., 2023), NLP for Islamic content is challenged by the limited availability of sizable datasets and constrained computational resources. Our work similarly faces these limitations, as it requires more extensive experimentation across a diverse range of LLMs to fully assess performance and robustness. Furthermore, the development of a reasonably sized, well-annotated dataset representative of the varied nature of Islamic texts would be instrumental in enabling more effective learning-based approaches. Such datasets could facilitate the use of smaller, more efficient LLMs to perform Islamic content processing and classical Arabic language tasks with higher accessibility and lower computational cost. Addressing these limitations remains an important direction for future research.

Acknowledgments

We sincerely thank the organizers for their dedication to advancing Islamic NLP and Information Retrieval research, especially in this groundbreaking era of powerful LLMs.

References

Saud Althabiti, Mohammad Ammar Alsalka, and Eric Atwell. 2024. Ta'keed: The first generative fact-checking system for arabic claims. *Preprint*, arXiv:2401.14067.

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.

Muhammad Huzaifa Bashir, Aqil M. Azmi, Haq Nawaz, Wajdi Zaghouani, Mona Diab, Ala Al-Fuqaha, and

Junaid Qadir. 2023. Arabic natural language processing for qur'anic research: a systematic review. *Artificial Intelligence Review*, 56(7):6801–6854.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

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.

Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing*, 8(4).

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, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. Hallucinations in large multilingual translation models. *Preprint*, arXiv:2303.16104.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

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 *Proceedinsg 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.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Darwish, and Walid Magdy. 2025. IslamicEval 2025: The First Shared Task of Capturing LLMs Hallucination in Islamic Content. In *Proceedings of ArabicNLP 2025*, TBD. Association for Computational Linguistics. To appear.

OpenAI team. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Marie C. Platenius, Markus von Detten, Steffen Becker, Wilhelm Schäfer, and Gregor Engels. 2013. A survey of fuzzy service matching approaches in the context of on-the-fly computing. In *Proceedings of the 16th International ACM Sigsoft Symposium on Component-Based Software Engineering*, CBSE '13, page 143–152, New York, NY, USA. Association for Computing Machinery.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

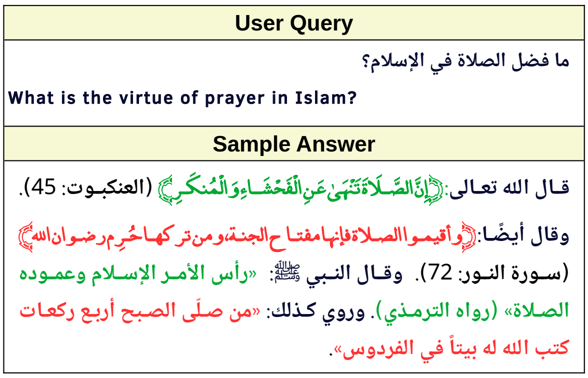
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Waleed A. Yousef, Taha M. Madbouly, Omar M. Ibrahime, Ali H. El-Kassas, Ali O. Hassan, and Abdallah R. Albohy. 2018. Pyquran: The python package for quranic analysis. https://hci-lab.github.io/PyQuran-Private.

Appendix

Figure 1 illustrates a sample hallucinated output generated by GPT-4o, demonstrating that even SOTA models can produce inaccurate Arabic Islamic content (Guerreiro et al., 2023).



User Query

ما فضل الصلاة في الإسلام؟

What is the virtue of prayer in Islam?

Sample Answer

قال الله تعالى: ﴿إِنَّ الصَّلَاةَ تَنْهَىٰ عَنِ الْفَحْشَاءِ وَالْمُنكَرِ﴾ (العنكبوت: 45).
وقال أيضًا: ﴿وأقيموا الصلاة فإنها مفتاح الجنة، ومن تركها حُرِم رضوان الله﴾
(سورة النور: 72). وقال النبي ﷺ: «رأس الأمر الإسلام وعموده
الصلاة» (رواه الترمذي). وروي كذلك: «من صلّى الصبح أربع ركعات
كتب الله له بيتاً في الفردوس».

Figure 1: Sample generated content by GPT-4o with color-coded verification: green indicates correct content, while red highlights invented Quran or Hadith. Some irrelevant content was truncated for clarity.

A Subtask 1A: Islamic Content Identification

A.1 Dataset Details

Table 4 presents the statistical analysis of the dataset for subtask **1A**. The dataset demonstrates varying annotation densities and imbalanced label distributions across the identification task.

A.2 System Design Details

Figure 2 provides an overall view of the system design for subtask **1A**. Algorithm 1 demonstrates the algorithmic pseudocode for the span extraction problem. Figure 3 shows the few-shot prompt template used for Islamic content identification.

Metric	Value
Unique Questions	50
Annotations per Question	4.20 ± 4.30
Ayahs per Question	2.36 ± 3.26
Hadiths per Question	1.52 ± 2.47
Label Distribution	
Ayah	118
Hadith	76
NoAnnotation	16

Table 4: Subtask **1A** dataset statistics: annotation density and class distribution for span extraction task.

B Subtask 1B: Islamic Content Verification

B.1 Dataset Details

Table 5 presents the statistical analysis of the dataset for subtask **1B**. The dataset demonstrates imbalanced label distributions across the binary classification verification task.

Metric	Value
Number of samples	247
Number of verses	4940
Number of unique questions	50
WrongAyah	70
CorrectAyah	110
WrongHadith	30
CorrectHadith	37

Table 5: Subtask **1B** dataset statistics: sample distribution and verification labels for binary classification task.

B.2 System Design Details

Figure 4 provides an overall view of the system design for subtask **1B**. Algorithm 2 demonstrates the algorithmic pseudocode for the verification problem. Figure 5 shows the few-shot prompt template used for binary classification in content verification.

B.3 Additional Results

Tables 6, 7, and 8 present comprehensive confusion matrices for different model configurations, evaluating performance across overall metrics, Quranic content verification, and Hadith content verification respectively.

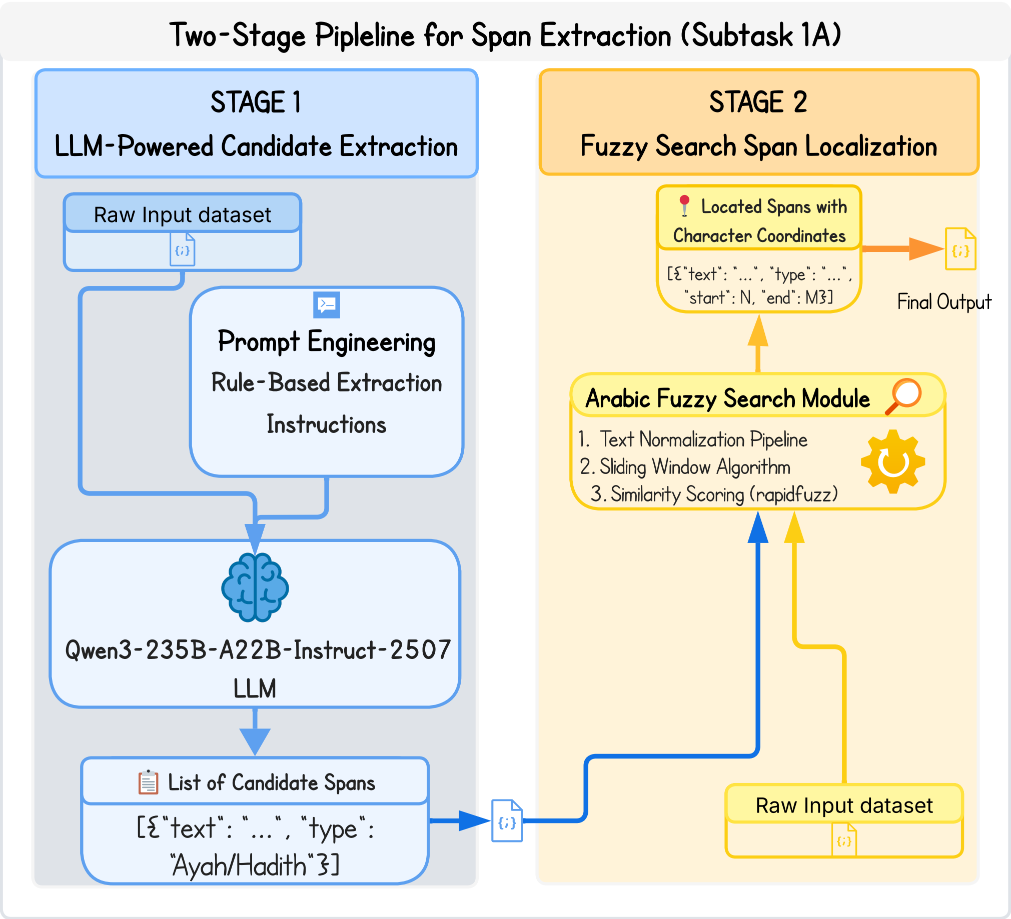
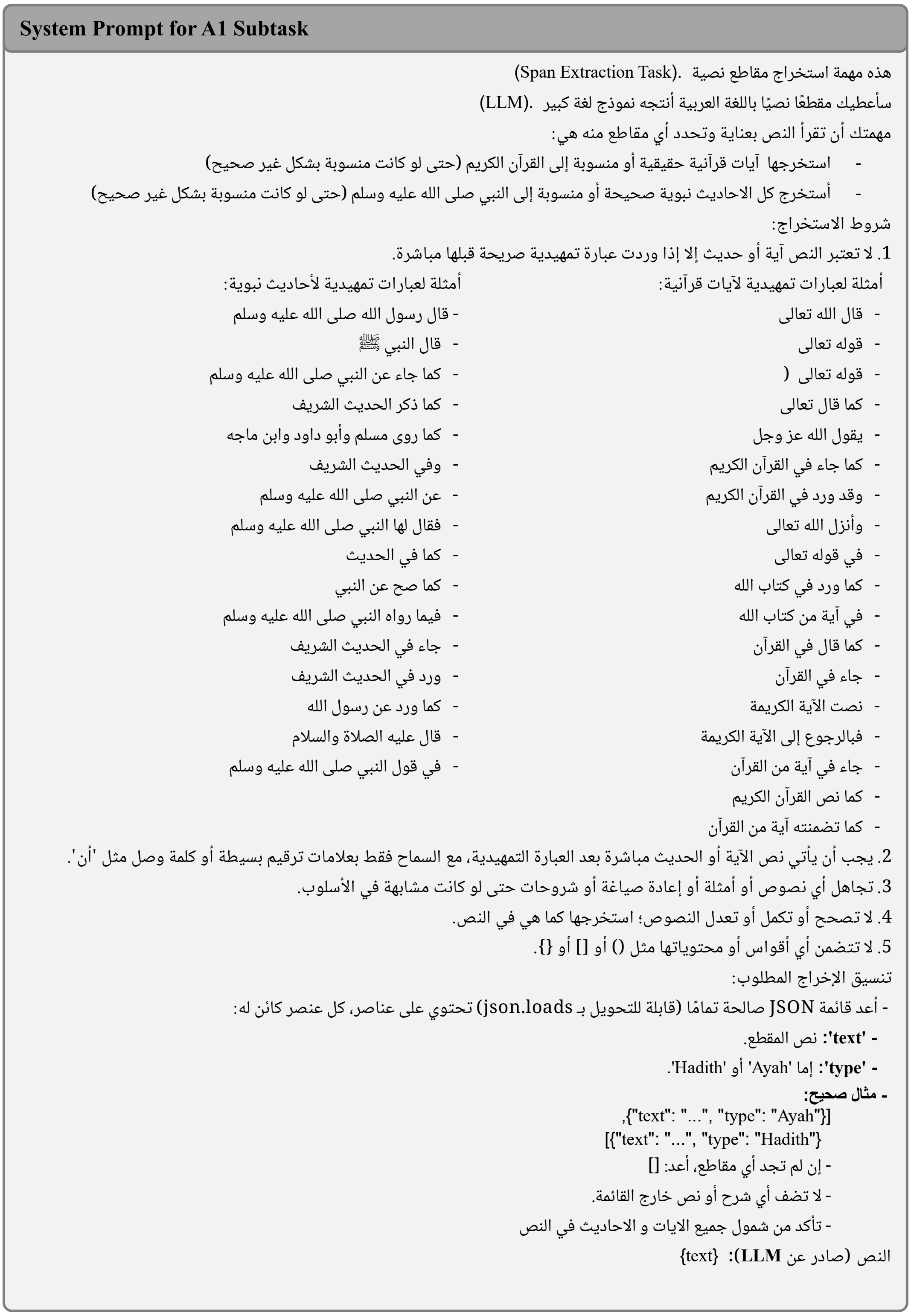

Two-Stage Pipleline for Span Extraction (Subtask 1A)
STAGE 1
LLM-Powered Candidate Extraction
Raw Input dataset
{;}
Prompt Engineering
Rule-Based Extraction
Instructions
Qwen3-235B-A22B-Instruct-2507
LLM
List of Candidate Spans
[{"text": "...", "type": "Ayah/Hadith"}]
{;}
STAGE 2
Fuzzy Search Span Localization
Located Spans with Character Coordinates
[{"text": "...", "type": "...", "start": N, "end": M}]
{;}
Final Output
Arabic Fuzzy Search Module
1. Text Normalization Pipeline
2. Sliding Window Algorithm
3. Similarity Scoring (rapidfuzz)
Raw Input dataset
{;}

Figure 2: Overall system architecture for Islamic content Identification (Subtask 1A).

Algorithm 1 Span Extraction with Fuzzy Matching

```
Require: Generated response text $T$, pretrained LLM, prompt template $P$, fuzzy matching threshold $\theta$
Ensure: Extracted and verified spans $S$
Define $\mathcal{F}(s, T)$ as fuzzy matching function returning set of matched entries with similarity scores
$T_{chunks} \leftarrow$ chunk text $T$ into manageable segments for LLM processing
Construct few-shot prompt $P$ emphasizing trigger words and citation patterns
$S_{\text{raw}} \leftarrow \emptyset$
for all chunk $c$ in $T_{chunks}$ do
    $S_{chunk} \leftarrow$ output spans extracted by LLM using prompt $P$ on chunk $c$
    $S_{\text{raw}} \leftarrow S_{\text{raw}} \cup S_{chunk}$
end for
$S \leftarrow \emptyset$
for all span $s$ in $S_{\text{raw}}$ do
    $M_s \leftarrow \mathcal{F}(s, T)$                                  ▷ Get matching results
    $m_s \leftarrow \max_{(e,\text{score}) \in M_s}$ score                  ▷ Select highest similarity score
    if $m_s \geq \theta$ then
        $S \leftarrow S \cup \{s\}$                                ▷ Add span to verified set
    end if
end for
return $S$
```



System Prompt for A1 Subtask

هذه مهمة استخراج مقاطع نصية .(Span Extraction Task)

سأعطيك مقطعًا نصيًا باللغة العربية أنتجه نموذج لغة كبير .(LLM)

مهمتك أن تقرأ النص بعناية وتحدد أي مقاطع منه هي:

- استخرجها آيات قرآنية حقيقية أو منسوبة إلى القرآن الكريم (حتى لو كانت منسوبة بشكل غير صحيح)
- أستخرج كل الاحاديث نبوية صحيحة أو منسوبة إلى النبي صلى الله عليه وسلم (حتى لو كانت منسوبة بشكل غير صحيح)

شروط الاستخراج:

1. لا تعتبر النص آية أو حديث إلا إذا وردت عبارة تمهيدية صريحة قبلها مباشرة.

أمثلة لعبارات تمهيدية لآيات قرآنية:

- قال الله تعالى
- قوله تعالى
- قوله تعالى (
- كما قال تعالى
- يقول الله عز وجل
- كما جاء في القرآن الكريم
- وقد ورد في القرآن الكريم
- وأنزل الله تعالى
- في قوله تعالى
- كما ورد في كتاب الله
- في آية من كتاب الله
- كما قال في القرآن
- جاء في القرآن
- نصت الآية الكريمة
- فبالرجوع إلى الآية الكريمة
- جاء في آية من القرآن
- كما نص القرآن الكريم
- كما تضمنته آية من القرآن

أمثلة لعبارات تمهيدية لأحاديث نبوية:

- قال رسول الله صلى الله عليه وسلم
- قال النبي ﷺ
- كما جاء عن النبي صلى الله عليه وسلم
- كما ذكر الحديث الشريف
- كما روى مسلم وأبو داود وابن ماجه
- وفي الحديث الشريف
- عن النبي صلى الله عليه وسلم
- فقال لها النبي صلى الله عليه وسلم
- كما في الحديث
- كما صح عن النبي
- فيما رواه النبي صلى الله عليه وسلم
- جاء في الحديث الشريف
- ورد في الحديث الشريف
- كما ورد عن رسول الله
- قال عليه الصلاة والسلام
- في قول النبي صلى الله عليه وسلم

2. يجب أن يأتي نص الآية أو الحديث مباشرة بعد العبارة التمهيدية، مع السماح فقط بعلامات ترقيم بسيطة أو كلمة وصل مثل 'أن'.

3. تجاهل أي نصوص أو أمثلة أو إعادة صياغة أو شروحات حتى لو كانت مشابهة في الأسلوب.

4. لا تصحح أو تكمل أو تعدل النصوص؛ استخرجها كما هي في النص.

5. لا تتضمن أي أقواس أو محتوياتها مثل () أو [] أو {}.

تنسيق الإخراج المطلوب:

- أعد قائمة JSON صالحة تمامًا (قابلة للتحويل بـ json.loads) تحتوي على عناصر، كل عنصر كائن له:

- **'text':** نص المقطع.

- **'type':** إما 'Ayah' أو 'Hadith'.

- مثال صحيح:

```
,{"text": "...", "type": "Ayah"}]
[{"text": "...", "type": "Hadith"}
```

- إن لم تجد أي مقاطع، أعد: []

- لا تضف أي شرح أو نص خارج القائمة.

- تأكد من شمول جميع الايات و الاحاديث في النص

النص (صادر عن **LLM**): {text}

Figure 3: Few-shot prompt template for span extraction in Subtask 1A: Islamic content identification using trigger words and citation patterns.

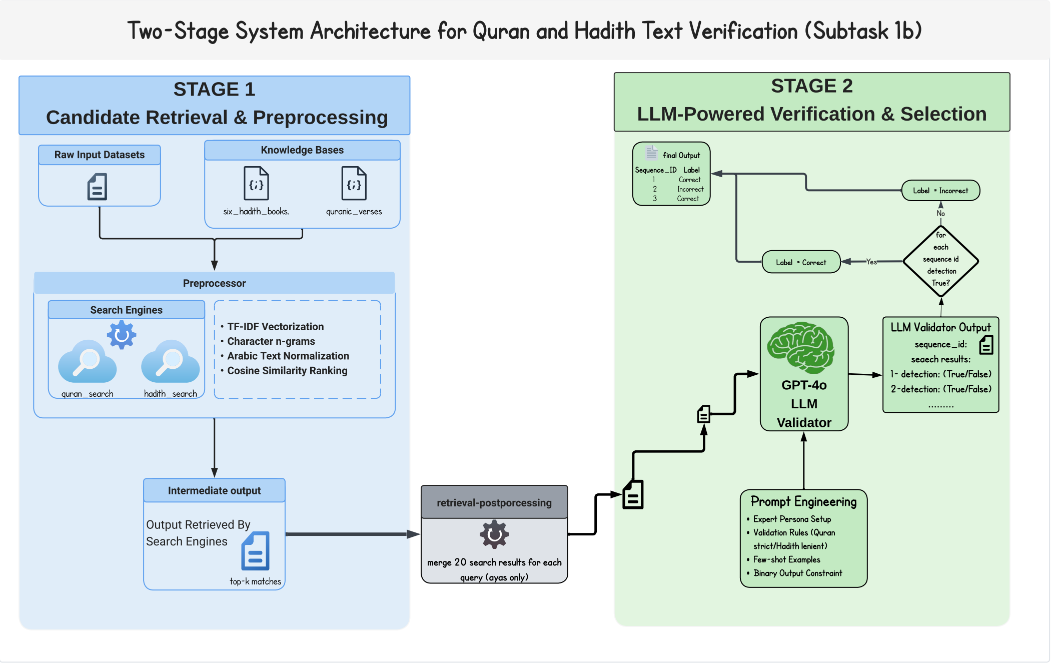
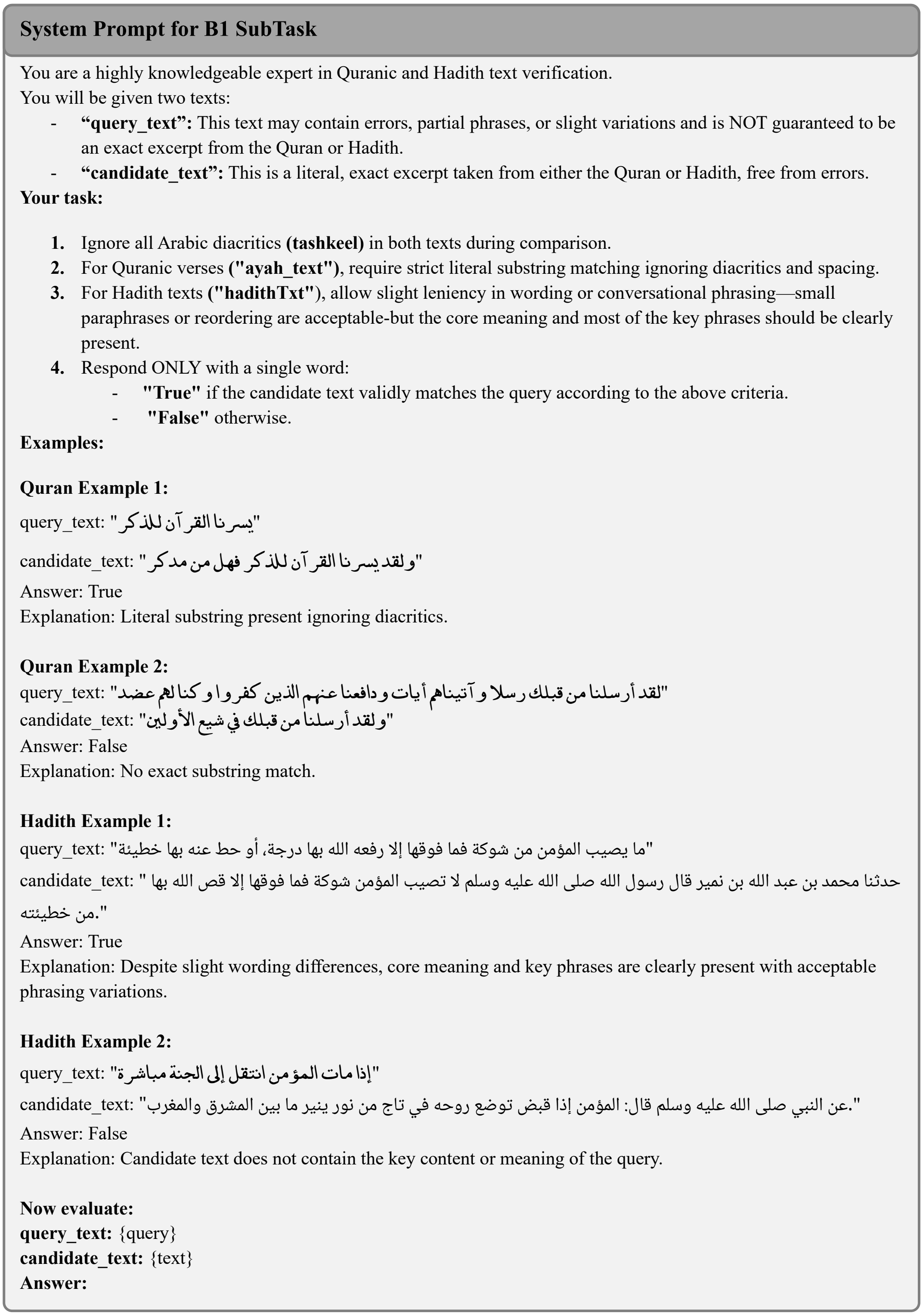


Figure 4: Overall system architecture for Islamic content Verification (Subtask 1B).

Algorithm 2 Verification with Hybrid Retrieval for Subtask 1B

```
Require: Extracted span $s$, Quranic database $DB_Q$, Hadith database $DB_H$, LLM, prompt templates
    $P_Q, P_H$, retrieval threshold $k$
Ensure: Verification result: Verified or Not Verified
Content type $t$ is provided from input file (Quranic or Hadith)
if $t$ is Quranic then
    Tokenize span $s$ into words $W = \{w_1, w_2, \ldots, w_n\}$
    $R \leftarrow$ retrieve top $k$ verses from $DB_Q$ using word-level voting
    $R_{merged} \leftarrow$ merge adjacent verses from same surah in $R$
    $P \leftarrow P_Q$                                        ▷ Strict word-for-word matching
else
    $R \leftarrow$ retrieve top $k$ Hadith entries from $DB_H$ using char-level TF-IDF
    $R_{merged} \leftarrow R$                                 ▷ No merging for Hadith
    $P \leftarrow P_H$                                        ▷ Allow minor variations
end if
for all retrieved result $r$ in $R_{merged}$ do
    result $\leftarrow$ LLM($P$, $s$, $r$)                    ▷ Few-shot binary classification
    if result is Verified then
        return Verified                                       ▷ Early termination
    end if
end for
return Not Verified
```



System Prompt for B1 SubTask

You are a highly knowledgeable expert in Quranic and Hadith text verification.
You will be given two texts:

- **“query_text”:** This text may contain errors, partial phrases, or slight variations and is NOT guaranteed to be an exact excerpt from the Quran or Hadith.
- **“candidate_text”:** This is a literal, exact excerpt taken from either the Quran or Hadith, free from errors.

Your task:

1. Ignore all Arabic diacritics **(tashkeel)** in both texts during comparison.
2. For Quranic verses **("ayah_text")**, require strict literal substring matching ignoring diacritics and spacing.
3. For Hadith texts **("hadithTxt"**), allow slight leniency in wording or conversational phrasing—small paraphrases or reordering are acceptable-but the core meaning and most of the key phrases should be clearly present.
4. Respond ONLY with a single word:
   - **"True"** if the candidate text validly matches the query according to the above criteria.
   - **"False"** otherwise.

Examples:

Quran Example 1:
query_text: "يسرنا القرآن للذكر"
candidate_text: "ولقد يسرنا القرآن للذكر فهل من مدكر"
Answer: True
Explanation: Literal substring present ignoring diacritics.

Quran Example 2:
query_text: "لقد أرسلنا من قبلك رسلا و آتيناهم أيات ودافعنا عنهم الذين كفروا و كنا لهم عضد"
candidate_text: "ولقد أرسلنا من قبلك في شيع الأولين"
Answer: False
Explanation: No exact substring match.

Hadith Example 1:
query_text: "ما يصيب المؤمن من شوكة فما فوقها إلا رفعه الله بها درجة، أو حط عنه بها خطيئة"
candidate_text: " حدثنا محمد بن عبد الله بن نمير قال رسول الله صلى الله عليه وسلم لا تصيب المؤمن شوكة فما فوقها إلا قص الله بها من خطيئته."
Answer: True
Explanation: Despite slight wording differences, core meaning and key phrases are clearly present with acceptable phrasing variations.

Hadith Example 2:
query_text: "إذا مات المؤمن انتقل إلى الجنة مباشرة"
candidate_text: "عن النبي صلى الله عليه وسلم قال: المؤمن إذا قبض توضع روحه في تاج من نور ينير ما بين المشرق والمغرب."
Answer: False
Explanation: Candidate text does not contain the key content or meaning of the query.

Now evaluate:
query_text: {query}
candidate_text: {text}
Answer:

Figure 5: Few-shot prompt template for binary classification in Subtask 1B: Quranic and Hadith content verification against authoritative sources.

Gemma-4B-IT (with diacritics)			**Gemma-4B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	144 (58.3%)	3 (1.2%)	Correct	145 (58.9%)	2 (0.8%)
Incorrect	*77 (31.2%)*	23 (9.3%)	Incorrect	*81 (32.9%)*	19 (7.7%)
Gemma-12B-IT (with diacritics)			**Gemma-12B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	145 (58.8%)	2 (0.8%)	Correct	145 (58.7%)	2 (0.8%)
Incorrect	*63 (25.5%)*	37 (15.0%)	Incorrect	*70 (28.3%)*	30 (12.1%)
GPT-4o (with diacritics)			**GPT-4o (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	145 (58.7%)	2 (0.81%)	Correct	145 (58.7%)	2 (0.81%)
Incorrect	*22 (8.9%)*	78 (31.57%)	Incorrect	*31 (12.5%)*	69 (27.93%)

Table 6: Confusion Matrices for Gemma and GPT Models (Overall Performance)

Gemma-4B-IT (with diacritics)			**Gemma-4B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	108 (60.0%)	2 (1.1%)	Correct	109 (60.6%)	1 (0.6%)
Incorrect	*57 (31.7%)*	13 (7.2%)	Incorrect	*62 (34.4%)*	8 (4.4%)
Gemma-12B-IT (with diacritics)			**Gemma-12B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	109 (60.6%)	1 (0.6%)	Correct	109 (60.6%)	1 (0.6%)
Incorrect	*49 (27.2%)*	21 (11.7%)	Incorrect	*52 (28.9%)*	18 (10.0%)
GPT-4o (with diacritics)			**GPT-4o (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	109 (60.6%)	1 (0.6%)	Correct	109 (60.5%)	1 (0.6%)
Incorrect	*19 (10.6%)*	51 (28.33%)	Incorrect	*28 (15%)*	42 (23.33%)

Table 7: Confusion Matrices for Gemma and GPT Models (Quranic Content)

Gemma-4B-IT (with diacritics)			**Gemma-4B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	36 (53.7%)	1 (1.5%)	Correct	36 (53.7%)	1 (1.5%)
Incorrect	*20 (29.9%)*	10 (14.9%)	Incorrect	*19 (28.4%)*	11 (16.4%)
Gemma-12B-IT (with diacritics)			**Gemma-12B-IT (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	36 (53.7%)	1 (1.5%)	Correct	36 (53.7%)	1 (1.5%)
Incorrect	*14 (20.9%)*	16 (23.9%)	Incorrect	*18 (26.9%)*	12 (17.9%)
GPT-4o (with diacritics)			**GPT-4o (no diacritics)**		
	Predicted			Predicted	
Actual	Correct	Incorrect	Actual	Correct	Incorrect
Correct	36 (53.7%)	1 (1.5%)	Correct	36 (53.7%)	1 (1.5%)
Incorrect	*3 (4.5%)*	27 (40.3%)	Incorrect	*3 (4.5%)*	27 (40.3%)

Table 8: Confusion Matrices for Gemma and GPT Models (Hadith Content)

Category	**Accuracy**	**Precision**	**Recall**	**F1 Score**
GPT-4o (no diacritics)				
Overall	86.6%	82.4%	98.6%	89.8%
Quran	83.9%	79.6%	99.1%	88.3%
Hadith	94.0%	92.3%	97.3%	94.7%
GPT-4o (with diacritics)				
Overall	90.3%	86.8%	98.6%	92.4%
Quran	88.9%	85.2%	99.1%	91.6%
Hadith	94.0%	92.3%	97.3%	94.7%

Table 9: GPT-4o Performance Metrics for Subtask 1B

ThinkDrill at IslamicEval 2025 Shared Task: LLM Hybrid Approach for Qur'an and Hadith Question Answering

Eman Elrefai[1]
[1]Alexandria University
eman.lotfy.elrefai@gmail.com

Toka Khaled[2]
[2]Al-Azhar University
Tokakhaled98@gmail.com

Ahmed Soliman[3]*
[3]University of Florida
ahmed.soliman@ufl.edu

Abstract

This paper presents our approach to Subtask 2 of IslamicEval 2025, a shared task that involves retrieving relevant passages from Quranic verses and Sahih Bukhari hadiths to answer Modern Standard Arabic (MSA) questions. We developed a multi-pipeline hybrid system that combines three complementary approaches: fine-tuned embedding models using triplet loss, keyword-based fuzzy matching, and large language model guided retrieval. Our system achieved `MAP_@10` of 0.2296, `MAP_Q@5` of 0.2623, and `MAP_H@5` of 0.215 in the test set, demonstrating the effectiveness of combining multiple retrieval strategies for Arabic religious text question answering.

1 Introduction

The Qur'an and Hadith Question Answering (QH-QA) task (Mubarak et al., 2025) addresses the challenge of retrieving relevant religious passages to answer questions posed in MSA. The Qur'an and hadith are deeply embedded in the daily lives of millions of Muslims worldwide, influencing their decisions, moral reasoning, and spiritual practices. With the increasing proliferation of Large Language Models (LLMs) in question-answering systems, systems responding to questions about these religious sources must maintain high accuracy and reliability.

This task builds on prior Qur'an QA challenges (2022, 2023) (Malhas et al., 2022, 2023), which focused only on Qur'an-based QA. Many teams proposed strong pipelines with promising results, and those works inspired our approach like Mahmoudi et al. (2023); Elkomy and Sarhan (2024). The main difference now is the inclusion of hadith, making the task broader and more challenging. Another key change is that answers must be retrieved from the entire Qur'an or hadith, unlike earlier setups where a specific passage was given and answers were extracted from it. Personally, our participation (Sleem et al., 2022) in Qur'an QA 2022 was a starting point that shaped how we combined prior pipelines with new technologies in this work.

Our main system strategy employs a multi-pipeline approach that leverages the strengths of different retrieval methods. Our key findings show that while individual approaches have limitations, their combination significantly improves performance. Our results show that the development set with `MAP_@10` of 0.32 and 0.2296 for the test set.

2 Background

The IslamicEval 2025 Subtask 2 requires systems to return a ranked list of answer-bearing passages from two collections: Quranic verses covering the Holy Qur'an and hadiths from Sahih Bukhari. Given a free-text question in MSA such as:

ما هو فضل ليلة القدر؟

The system should return relevant passages like:

إِنَّا أَنزَلْنَاهُ فِي لَيْلَةِ الْقَدْرِ * وَمَا أَدْرَاكَ مَا لَيْلَةُ الْقَدْرِ * لَيْلَةُ الْقَدْرِ
خَيْرٌ مِّنْ أَلْفِ شَهْرٍ (Rank 1)
عَنْ أَبِي هُرَيْرَةَ رَضِيَ اللَّهُ عَنْهُ قَالَ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ
عَلَيْهِ وَسَلَّمَ: مَنْ قَامَ لَيْلَةَ الْقَدْرِ إِيمَانًا وَاحْتِسَابًا غُفِرَ لَهُ مَا تَقَدَّمَ
مِنْ ذَنْبِهِ (Rank 2)

2.1 Dataset Details

The dataset consists of 1,266 Quranic passages from the Quranic Passage Collection (QPC), 2,254 hadiths from Sahih Bukhari, training questions with manually annotated relevance judgments, and questions without answers marked with passage ID "-1". Initially, the training data only contained Qur'an answers. Our team manually added hadith answers to create a more balanced training set.

[*] Also affiliated with Al-Azhar University

Proceedings of The Third Arabic Natural Language Processing Conference, pages 528–533
November 8-9, 2025 ©2025 Association for Computational Linguistics

2.2 Related Work

Previous work on Arabic question answering has primarily focused on general domain texts. Earlier versions of similar tasks focused exclusively on Quranic sources, but the inclusion of hadith as a complementary resource introduces additional complexity. Hadith collections present unique challenges due to their narrative structure, chain of transmission (isnad), and the potential for fabrication, requiring careful verification and authentic sourcing.

While several retrieval systems have been developed specifically for hadith collections (Mahmood et al., 2018), fewer systems effectively combine Qur'an and hadith sources in a unified retrieval framework. Recent work in (Fawzi et al., 2025) demonstrates the importance of accurate religious text retrieval systems, particularly given the widespread influence of these sources on personal decisions and the need for reliable information retrieval in the era of increasing LLM deployment.

Our approach builds upon sentence transformers for multilingual retrieval while addressing the specific requirements of Islamic texts and the challenge of combining these two distinct yet complementary religious sources.

3 System Overview

Our hybrid system consists of three complementary pipelines designed to capture different aspects of semantic similarity and relevance. Figure 1 illustrates the overall architecture of our multi-pipeline approach.

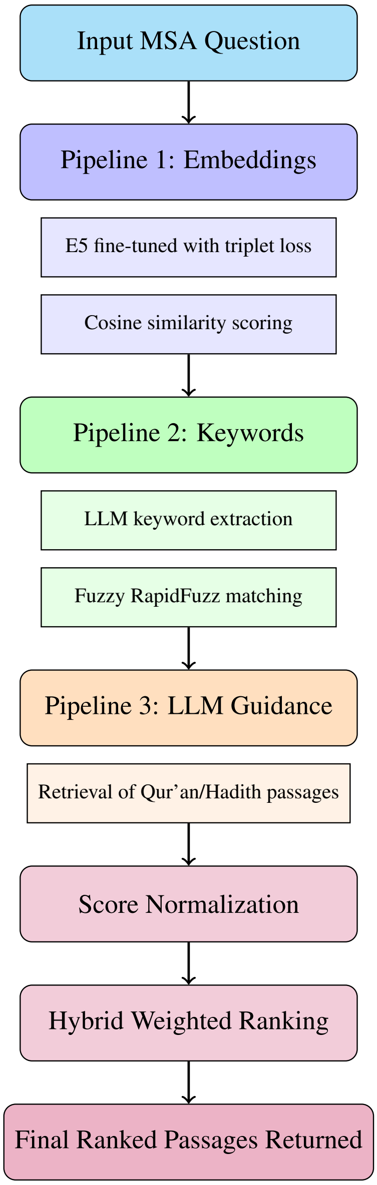


Figure 1: System architecture: input questions are processed via three pipelines with distinct colors.

3.1 Pipeline 1: Fine-tuned Embedding Model

3.1.1 Training Phase

The training pipeline relied on a curated dataset constructed from multiple sources to ensure comprehensive coverage of Qur'anic and Hadith material. First, official Qur'an QA pairs provided by the competition were used as a foundation. To expand beyond the Qur'an, additional Hadith QA pairs were constructed by sourcing relevant narrations from *Sahih al-Bukhari*. This was feasible only for a limited subset of questions, so we further incorporated the HAQA dataset, aligning its QA pairs with *Sahih al-Bukhari* narrations through automated normalization (removing diacritics, punctuation, and text inconsistencies) and fuzzy matching. Matches with similarity scores above a chosen threshold were retained, producing a final aligned dataset containing question text, answer, and narration. The HAQA dataset is available at `https://github.com/scsaln/HAQA-and-QUQA/blob/main/HAQA.csv`. This curated dataset balanced Qur'anic and Hadith sources, enabling the retrieval model to learn cross-domain semantic relationships.

On top of this dataset, we fine-tuned a multilingual sentence transformer (Reimers and Gurevych, 2019) using triplet loss (Yeruva et al., 2022). The augmentation process expanded the original corpus into structured triplets by systematically constructing positive and negative passages for each question:

- **Positive passages:** For each question, all valid answers from the Qur'an and Hadith were included as positives. When multiple passages addressed the same question, each of them was considered a valid positive. For unanswerable questions, we used the placeholder answer لا يوجد as the only positive.

- **Negative passages:** Non-relevant passages were sampled from the remaining pool of Qur'an and Hadith texts. For unanswerable questions, all real passages in the corpus were treated as negatives.

- **Triplet construction:** Each training instance consisted of an anchor (the question), a positive passage, and a negative passage. To increase data diversity, multiple triplets were generated per question by pairing the same anchor with different positive and negative samples.

The fine-tuned model based on `intfloat/multilingual-e5-base` served as the **retriever**, encoding both queries and passages into a shared embedding space and retrieving candidate passages using cosine similarity. To further refine the retrieval results, we employed the **reranker** model, specifically the pretrained `cross-encoder/ms-marco-MiniLM-L-6-v2`, which jointly encodes query–passage pairs and assigns a relevance score. This two-stage pipeline ensured efficient large-scale retrieval while improving precision through reranking.

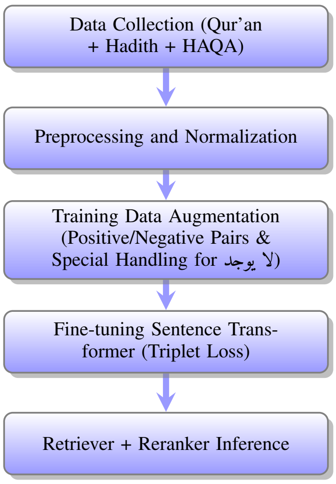

Data Collection (Qur'an + Hadith + HAQA)
Preprocessing and Normalization
Training Data Augmentation (Positive/Negative Pairs & Special Handling for لا يوجد)
Fine-tuning Sentence Transformer (Triplet Loss)
Retriever + Reranker Inference

Figure 2: Pipeline for fine-tuning and retrieval

3.1.2 Inference Phase:

Once the model is fine-tuned, it is employed in a retrieval pipeline for inference. Queries are encoded into embeddings and matched against a vector database of Qur'an and Hadith passages. A retriever retrieves the top candidate passages using cosine similarity, which are then refined by a reranker before producing the final ranked results. Unlike full retrieval-augmented generation (RAG) systems, our pipeline focuses solely on retrieving and ranking authoritative passages without generating new text.

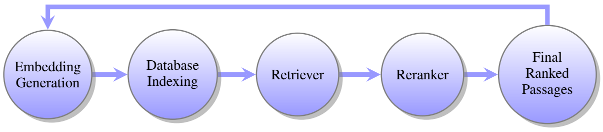


Figure 3: Retrieval and reranking pipeline

3.2 Pipeline 2: Keyword-based Fuzzy Matching

We used GPT-4 to extract relevant keywords from the questions and then used fuzzy string matching to find passages containing similar terms. We used RapidFuzz (Ye et al., 2021) a fast Python library for fuzzy string matching to compute partial ratio similarity scores. The algorithm extracts keywords using the LLM prompt "Give me the main keywords that I can search for to get answers from the Qur'an and Hadith", cleans the Arabic text by removing diacritics and normalising the characters, applies fuzzy partial ratio matching with a threshold of 70%, and ranks results by similarity score. This approach complements semantic matching by capturing cases where wording is very similar but embeddings may miss exact phrasing.

3.3 Pipeline 3: LLM-guided Retrieval

The input to this pipeline is the users question together with the instruction: "Answer questions using only Quran and Sahih Bukhari. Provide exact verses/hadiths, not interpretations. Use -1 if no answer exists."

The output is either the exact verse or hadith matching the question, or -1 if no relevant answer is found.

We chose Claude Sonnet 4 because it follows instructions well, handles long passages reliably, and shows fewer hallucinations than smaller or larger alternatives. It also provides a good balance between accuracy, speed, and cost.

3.4 Hybrid Combination

Results from all three pipelines were combined using score normalization and weighted averaging to produce final rankings.

4 Experimental Setup

4.1 Data Preparation

Following cleaning, the dataset was structured for Sentence-BERT (SBERT) triplet loss training. Triplet construction formatted each entry as (anchor, positive, negative), where anchor represents the text of the question, positive encompasses corresponding relevant answers from the Qur'an or hadith, and negative includes semantically irrelevant passages from the Qur'an or hadith. Data splitting used stratified sampling to ensure both Quranic and hadith entries were proportionally represented in training and validation sets. UTF-8 encoding stored all text fields in a Pandas DataFrame with explicit column names (`question`, `positive_passage`, `negative_passage`).

4.2 Data Preprocessing

We applied preprocessing to align with sentence transformer requirements. Data was length-filtered (10512 tokens) and segmented using a sliding window to preserve context within token limits. Arabic-specific cleaning included diacritic removal, normalization of letter variants, tatweel and honorific symbol removal, and whitespace normalization. Stopwords were retained due to their semantic role in Quranic and hadith texts, while redundant punctuation was removed. Texts were tokenized with the `intfloat/multilingual-e5-base` (Wang et al., 2024) tokenizer, and triplets were batched into uniform tensors with attention masks for SBERT triplet loss training. For long passages, we applied chunking into 150-character segments with 30-character overlap to enhance retrieval granularity.

4.3 Training Configuration

We used base model `intfloat/multilingual-e5-base`, 2 epochs, batch size 16 with gradient accumulation, learning rate 2e-5 with 100 warm-up steps, triplet loss function with cosine distance, and hardware acceleration through Google Colab with GPU.

4.4 Evaluation Metrics

The official metrics included `MAP@10` (Mean Average Precision at rank 10), `MAP_Q@5` (MAP at rank 5 for Qur'an passages only), and `MAP_H@5` (MAP at rank 5 for hadith passages only).

5 Results and Error Analysis

Our system was evaluated on both development and test sets, achieving the following results:

Dataset	MAP@10	MAP_Q@5	MAP_H@5
Development	0.32	0.35	-
Test	0.2296	0.2623	0.215

Table 1: Overall performance on development and test sets.

To better understand these results, we further analyzed the contribution of each pipeline component. Since the organizers provided official test set results only for the submitted runs, the per-pipeline results in Table 2 were computed on the development set using the released evaluation script. The Hybrid Combination score corresponds to our submitted run on the test set. Table 2 reports the performance of individual pipelines compared to the hybrid system.

Pipeline	MAP@10
Embedding Model Only	0.15
Keyword Matching Only	0.08
LLM-guided Only	0.12
Hybrid Combination	0.173

Table 2: Performance of individual pipelines on the development set.

The fine-tuned embedding model provided the strongest standalone baseline, while keyword matching proved useful for questions relying on exact term overlap. The LLM-guided approach showed potential but was constrained by input length limitations. The hybrid combination achieved the best balance, outperforming any individual pipeline.

5.1 Coverage Analysis

On the test set of 71 questions, our retrieval system achieved 76.1% coverage: 54 questions had answers while 17 questions were marked as "no answer." On average, the system returned 15.2 passages per question.

5.2 Error Analysis

We observed four main error types: semantic mismatch (retrieving passages with overlapping words but different intent, e.g., prayer times vs. prayer importance), keyword limitations (lexical matches missing conceptual meaning), LLM constraints (token limits restricting comprehensive answers), and domain specificity (questions requiring advanced theological knowledge). Example errors include ما عمر الأرض بالسنوات تحديدًا؟ (expected: -1, predicted: creation verses)

5.3 No-Answer Detection

We evaluated the system's ability to detect questions without valid answers. A confidence threshold of 0.35 was applied: if the highest passage score fell below this threshold, the system classified the question as no answer. Evaluation was carried out on the held-out test set of 71 questions, which included 17 questions without valid answers. The model achieved a precision of 0.65 and recall of 0.47 on this subset.

6 Conclusion

We proposed a hybrid approach for Arabic Qur'an and hadith question answering that integrates fine-tuned embeddings, keyword matching, and LLM guidance. Our system demonstrated strong performance during development (`MAP@10`: 0.32) and achieved one of the top scores among participating teams on the final benchmark (`MAP@10`: 0.173). These results highlight both the effectiveness of our design and the potential for further improvements in handling diverse real-world queries.

Future work directions include incorporating Islamic scholarly knowledge graphs, exploring retrieval-augmented generation approaches, and creating larger, more diverse training datasets with theological expert annotations. The task highlights the complexity of understanding religious texts and the need for specialized approaches beyond general-domain techniques.

For reproducibility, the implementation and code are available at `ThinkDrill at IslamicEval 2025 | GitHub`.

Acknowledgments

We thank the IslamicEval 2025 shared task organizers for providing this valuable evaluation framework. We also acknowledge the anonymous reviewers for their constructive feedback.

References

Mohammed Alaa Elkomy and Amany Sarhan. 2024. Tce at qur'an qa 2023 shared task: Low resource enhanced transformer-based ensemble approach for qur'anic qa. *arXiv preprint arXiv:2401.13060*.

Mahmoud Fawzi, Walid Magdy, and Björn Ross. 2025. 'the prophet said so!': On exploring hadith presence on arabic social media. *Proceedings of the ACM on Human-Computer Interaction*, 9(2):1–23.

Ahsan Mahmood, Hikmat Ullah Khan, Fawaz K Alarfaj, Muhammad Ramzan, and Mahwish Ilyas. 2018. A multilingual datasets repository of the hadith content. *International Journal of Advanced Computer Science and Applications*, 9(2):165–172.

Ghazaleh Mahmoudi, Yeganeh Morshedzadeh, and Sauleh Eetemadi. 2023. Gym at quran qa 2023 shared task: Multi-task transfer learning for quranic passage retrieval and question answering with large language models. In *Proceedings of ArabicNLP 2023*, pages 714–719.

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 *Proceedinsg 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.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ahmed Sleem, Eman Mohammed lotfy Elrefai, Marwa Mohammed Matar, and Haq Nawaz. 2022. Stars at quran qa 2022: Building automatic extractive question answering systems for the holy quran with transformer models and releasing a new dataset. In *Proceedinsg 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 146–153.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Aoshuang Ye, Lina Wang, Lei Zhao, Jianpeng Ke, Wenqi Wang, and Qinliang Liu. 2021. Rapidfuzz: Accelerating fuzzing via generative adversarial networks. *Neurocomputing*, 460:195–204.

Nagamani Yeruva, Sarada Venna, Hemalatha Indukuri, and Mounika Marreddy. 2022. Triplet loss based siamese networks for automatic short answer grading. In *Proceedings of the 14th annual meeting of the forum for information retrieval evaluation*, pages 60–64.

Burhan at IslamicEval: Fact-Augmented and LLM-Driven Retrieval for Islamic QA

Mohammad Basheer Kotit[1] , **Watheq Mansour**[2], **Abdulhamid Touma**[3], **Ahmad Qadeib Alban**[4]

[1] Qatar University, Qatar
[2] The University of Queensland, Australia
[3] Syrian Virtual University, Syria
[4] Université Paris, Dauphine-PSL, France
mk1806194@qu.edu.qa, w.mansour@uq.edu.au, abdulhamid_103004@svuonline.org, Ahmad.qadeib-alban@dauphine.psl.eu

Abstract

This paper presents our approach to the Qur'an and Hadith QA task in the IslamicEval 2025 Shared Task. Reliable retrieval requires both accuracy and context-aware answers from Qur'anic and Hadith text. To address this challenge, We combine semantic search with LLM-based re-ranking. To enhance alignment, we augment the corpus with LLM-extracted Islamic facts and paraphrased queries. An LLM-based binary classifier further verifies whether retrieved passages answer the questions. Results show improved accuracy and better alignment with user intent.

1 Introduction

The Holy Qur'an, a sacred and timeless text revealed over 1400 years ago in Classical Arabic, continues to attract the attention of millions of Muslims and non-Muslims for its profound teachings, legislation, and extensive body of knowledge. Therefore, developing effective systems for the Holy Qur'an, particularly for Passage Retrieval (i.e., the task of identifying and ranking candidate passages that potentially contain answers to a given question), have become a matter of paramount importance (Malhas et al., 2022) and presents unique and significant challenges (Malhas et al., 2022; Zekiye and Amroush, 2023). The challenges stem from linguistic complexities, context scarcity, and the reliability and specificity required in the Holy Qur'an. Recently, the Qur'an QA 2023 shared task dataset has further highlighted the complexity of this task, and the results revealed substantial space for further improvements (Basem et al., 2024). In continuation of this effort, Qur'an and Hadith QA 2025 is offered as subtask in IslamicEval shared task (Mubarak et al., 2025). The main difference of this year task is the addition of Hadith (Sahih Bukhari collection, in particular) to the retrieval collections, making the task more challenging.

In this paper, we present our participation in Qur'an and Hadith QA 2025 subtask and describe our proposed retrieval pipeline. The main characteristics of our system are: **(1) Augment the Qur'an and Hadith** collections with information extracted by large language models (LLMs). **(2) Employ semantic search** to form the initial retrieval list, followed by LLM-based re-ranker to prioritize the most relevant candidates. **(3) Paraphrase user queries** using LLMs to enhance semantic clarity and improve retrieval outcomes. **(4) Employ a LLM-based binary classifier** to detect questions with no answers.

1.1 Related Work

The task of question answering (QA) for the Holy Qur'an was introduced as a shared task in (Malhas et al., 2022). The following year, the first task of Qur'anic passage retrieval was offered as a shared task (Malhas et al., 2023). The task of Qur'anic passage retrieval has garnered significant scholarly interest due to the distinct linguistic and contextual challenges posed by the Qur'an (Basem et al., 2024). Effective systems must retrieve relevant verses to answer both factoid and non-factoid questions and bridge the linguistic gap between Modern Standard Arabic (MSA) and Classical Arabic. A further challenge lies in detecting zero-answer scenarios, where questions that have no answers within the Qur'anic passages require robust mechanisms for rejecting all the non-relevant candidates (Malhas et al., 2023).

Several teams participated in the task and employed various technique such as augmentation (Elkomy and Sarhan, 2024; Basem et al., 2024), translation (Alawwad et al., 2023). As the augmentation showed noticeable improvements, we decide to continue exploring in this direction and propose new augmentation techniques.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 534–539
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

In this section, we present the required background information about the shared task.

2.1 Task Definition

Our work was merely on IslamicEval Subtask 2: Qur'an and Hadith QA 2025, which is a retrieval task and a continuation of Qur'an QA 2022 and Qur'an QA 2023 shared tasks. The task is defined as follows: Given a free-text question posed in MSA, a collection of Qur'anic passages (that cover the Holy Qur'an), and a collection of Hadith from Sahih Bukhari, a system is required to retrieve a ranked list up to 20 answer-bearing Qur'anic passages or Hadith (i.e., Islamic sources that potentially enclose the answer(s) to the given question) from the two collections. The question can be a factoid or a non-factoid question. To make the task more challenging, the organizers add on purpose some questions that have no answers in the Holy Qur'an, Sahih Al-Bukhari, or both. For such cases, the ideal system should return no answers.

2.2 Dataset Details

The dataset proposed for the task comprises two collections: the Qur'anic Passage Collection (QPC) and the Sahih Al-Bukhari Collection (SBC) [1]. The QPC segments the 114 chapters of the Qur'an into 1,266 topical passages, while the Sahih Al-Bukhari Collection includes 2,254 Hadith. To enable training, the organizers provide 250 questions of the AyaTEC dataset along with their relevance judgments over the Qur'anic Passage collection only. The questions are divided into training (84%) and development (16%) datasets.

3 System Overview

In this section, we illustrate the proposed retrieval pipeline to address the task at hand.

3.1 Data Augmentation and Information Extraction

Our augmentation strategy involves two approaches. In the first approach, following Elkomy and Sarhan (2024), we utilize two Tafsir sources (Al-Tafsir Al-Muyassar and Tafsir Al-Jalalayn) to augment the QPC passages with relevant interpretations. We believe this step is helpful in expanding the context as the text in QPC is generally short.

In the second approach, we extract factual information from QPC and SBC passages and then append them to the original text. The intuition behind this is that many MSA questions differ linguistically from the original Qur'anic or Hadith wording, requiring a deep semantic understanding of the content. We address this need by enriching QPC and SBC through extracting explicit semantic representations using LLMs. By generating explicit facts, we bridge the linguistic gap, thereby improving semantic search recall. To achieve this goal, we develop a domain-adapted prompt (Figures 1–2) to extract key entities and relations —characters, places, events, Islamic concepts, and legal rulings— from each Qur'anic passage or Hadith. In response to our prompts, the LLM rewrites the implicit references into unambiguous descriptions, enabling the retrieval model to better align them with the query's intent. In other words, the proposed method captures both explicit and implicit meanings that standard embedding models may not explicitly state in the surface text. Finally, we pair the extracted information (IE) with its corresponding text. We refer to this pairing by QPC + IE and SBC + IE for the Qur'an and the Hadith, respectively.

3.2 Semantic Retrieval with LLM Re-ranking

We generate embeddings for all enriched texts in QPC and SBC using a SOTA embedding model ("text-embedding-3-large" [2]), as explained in appendix A. Query embeddings are also generated using the same model to ensure proper semantic matching.

Following this, we apply semantic search independently for each source (QPC and SBC). In the initial retrieval phase, cosine similarity is computed between the query embedding and the (QPC or SBC) embeddings. For each collection, the top 20 passages are selected, and then the two lists are merged to form the initial retrieval set (referred to Dense).

To enhance the retrieval quality, we introduce a reranking stage by pairing the candidate passages from the initial retrieval set with the user query and pass them to an LLM. The LLM, in turn, reorders the retrieved passages according to their estimated relevance to the query.

[1] https://gitlab.com/bigirqu/quran-hadith-qa-2025

[2] https://platform.openai.com

3.3 Query Rewriting

Towards enhancing the retrieval performance, we utilized two methods to change the query text. In the first method, we augment the query with synonym words, and in the second one, we rephrase the query to a new form. We generate a new query file for each method and per a different LLM. Following this, we feed the generated query file to the ranking pipeline described in the previous steps.

Synonym Expansion. Since our retrieval pipeline is mainly focused on semantic matching, we believe that adding some synonym words to the query might increase the query-passage matching. Therefor, we employ LLMs to generate one synonym word for a list of query words. The list of query words is formed after removing the Arabic stop words, such as "What", and "Why", etc. Then, each generated word is positioned after its corresponding synonym between two parentheses. Here is an example of an expanded query:

**من هو النبي (الرسول) المعروف (المشهور)
بالصبر (بالتحمل) ؟**

Query Rephrasing. As LLMs are powerful writers, we decided to use their potential in paraphrasing the input query. Simply, we prompt an LLM to rewrite the given question in a better way.

3.4 LLM-Based No-Answer Detection

Following the reranking stage, we further refine the reranking set of passages to determine whether it contains an answer to the query. In particular, we prompt an LLM to make a binary judgment on a question-passage pair, assessing whether a given passage addresses the question explicitly or implicitly. If none of the passages in the reranked set are judged relevant, the system returns a standardized no-answer response; otherwise, the reranked list is preserved in its order.

Dataset	MAP@10	MAP@5
QPC	0.2761	0.2553
QPC + jalalayn	0.2798	0.2572
QPC + muyassar	0.2926	**0.2708**
QPC + IE	**0.2944**	0.2689
QPC + muyassar + IE	0.2878	0.2662

Table 1: Effect of augmenting QPC on semantic search on both the train and dev sets.

4 Experimental Setup

In the data augmentation phase, we used OpenAI's GPT-4o to extract factual statements from each passage in both QPC and SBC. To identify the most effective embedding model, we evaluated several Arabic embedding models, as detailed in appendix A. The "text-embedding-3-large" model demonstrated the highest overall performance and was therefore used in all subsequent experiments. Document embeddings were stored in ChromaDB [3], a persistent vector store, with cosine similarity as the distance metric.

Retrieval was conducted independently for the QPC and SBC datasets. For each collection, we retrieved the top 20 passages based on cosine similarity between the query and the passage embeddings, resulting in a combined list of 40 candidate passages. These were then reranked using OpenAI GPT-4o. For query paraphrasing, we test three variants of OpenAI GPT-4, namely: GPT-4o, GPT-4.1-mini, and GPT-4.1, selecting the latter for subsequent experiments due to its superior performance. Additionally, GPT-4o was employed as a binary classifier to determine whether each candidate passage was relevant to a given query.

Evaluation We report our evaluation result on a set *combined from the training and development* sets as we believe this gives more reliable and robust results compared to dev set only. While, we report MAP@5 and MAP@10 on the combined set, we report MAP@10, MAP_Q@5, and MAP_H@5 on the test set (as provided by the organizers).

Model	MAP@10	MAP@5
-	0.2944	0.2689
Query + Synonyms		
GPT-4.1-mini	0.2691	0.2453
GPT-4.1	0.2754	0.2540
GPT-4o	0.2781	0.2560
Paraphrased Query		
GPT-4.1-mini	0.3007	0.2727
GPT-4.1	**0.3065**	**0.2815**
GPT-4o	0.3026	0.28

Table 2: Results of different query expansion techniques using QPC+IE on both the train and dev sets.

[3] https://www.trychroma.com/

Method	MAP@10	MAP@5
PQ + Dense	0.3065	0.2815
PQ + Dense + RR_{CE} (Elkomy and Sarhan, 2024)	0.3156	0.2905
PQ + Dense + $RR_{\text{GPT-4}}$	0.4079	0.3898
PQ + Dense + $RR_{\text{GPT-4}}$ + NAD	0.4682	0.4511
Dense + $RR_{\text{GPT-4}}$ + NAD	**0.4811**	**0.4660**

Table 3: Effect of introducing reranker (RR) and no-answer detector (NAD) on performance using the paraphrased query (PQ) and augmented corpus (QPC + IE) on both the train and dev sets. GPT-4 and CE refers to GPT-4 and cross-encoder-based rerankers, respectively.

Method	Collection	MAP@10	MAP_Q@5	MAP_H@5
PQ + Dense + $RR_{\text{GPT-4}}$ + NAD	QPC + IE & SBC+ IE	**0.3351**	**0.3389**	**0.3876**
Dense + $RR_{\text{GPT-4}}$ + NAD	QPC + IE & SBC+ IE	0.3021	0.3091	0.3461
Dense + $RR_{\text{GPT-4}}$ + NAD	QPC + IE & SBC	0.2916	0.3130	0.2936

Table 4: Performance of retrieval strategies for related QPCs and HAs given a query on the test set.

5 Results and Analysis

In this section, we present the research questions along with the experiments that answer them.

RQ1: How does augmenting QPCs affect semantic retrieval?

In Table 1, we present the evaluation results of the two proposed augmentation approaches (with Tafsir and with facts extracted by LLMs (IE)). It is evident that augmenting QPC provides complementary semantic signals. Notably, combining QPC with either Muyassar or IE leads to observable performance gains, confirming the benefit of pairing verse-level content with simplified or pedagogically aligned annotations. However, adding Muyassar and IE together leads to a decline in the performance. We attribute this to the semantic noise when too many interpretative strategies are combined, potentially reducing coherence in the learned embedding space. To this end, we adopt SBC + IE in subsequent experiments.

RQ2: How effective are LLMs in reformulating the user queries?

In Table 2, we examine the effect of introducing the paraphrasing and adding synonyms to queries using three variants of GPT-4. The results reveal a clear distinction between the effectiveness of synonym-based and paraphrased query reformulations in Quranic semantic search. Synonym-based reformulations consistently underperformed the baseline, indicating that direct lexical substitution introduces noise and query drift. In contrast, paraphrased queries yield consistent improvements across all evaluated models. These gains highlight the strength of paraphrasing in capturing deeper semantic equivalences and aligning user queries more effectively with relevant passages. We select GPT-4.1 for paraphrasing queries in the following experiments due to its superior performance.

RQ3: How good is the LLM-based reranker? What is the best combinations of our proposed retrieval pipeline?

Building on the best results attained from augmentation and paraphrasing queries (PQ), we examine the effect of incorporating the reranker. In Table 3, we demonstrate the effectiveness of using a finetuned cross-encoder(CE)-based (Elkomy and Sarhan, 2024) and GPT-4-based rerankers. While CE-based reranker leads to moderate improvements, a substantial gain is achieved by the LLM-based re-ranker (0.4079 vs. 0.3156 at MAP@10).

In the same table, we report the significant gains brought by integrating the NAD (No-Answer Detection) component, which filters out candidates that do not answer the query.

Building on these findings, we submit the top-two performing pipelines (last two lines of Table 3) on the test set, while adding another run without augmenting SBC to test its effect. The results on the test set are shown in Table 4. The best results are obtained when PQ is combined with the Dense + RR + NAD pipeline on the augmented collection (QPC + IE & SBC + IE), indicating the effectiveness of PQ component.

6 Conclusion

In this paper, we described our method for addressing Qur'an and Hadith QA 2025 shared task. We found out that the augmenting QPC and SBC with information extracted by LLM lead to remarkable gains. After experimenting with multiple lexical- and semantic-based retrieval and reranking methods, we showed that dense search with LLM-based reranker is the best configuration. Our novel attempt to change the query surface text showed clear improvements. Finally, utilizing LLM to judge the binary relevance of a query-passage pair proved to be a promising solution in detecting questions with no answer.

Limitations

To the best of our knowledge, resources providing tafsir or detailed explanations of Hadith from Sahih al-Bukhari are not readily available.

In addition, our preliminary experiments were conducted exclusively on the QPC dataset, as it is the only resource with ground truth annotations available. Consequently, we did not develop or evaluate our best-performing retrieval system for retrieving passages from QPC or Hadith collections. Accordingly, the findings reported at this stage are limited in scope, which reduces confidence in identifying the optimal strategy.

Acknowledgments

We thank Dr. Joel Mackenzie for the resource support provided to Watheq Mansour.

References

Hessa Alawwad, Lujain Alawwad, Jamilah Alharbi, and Abdullah Alharbi. 2023. Ahjl at qur'an qa 2023 shared task: Enhancing passage retrieval using sentence transformer and translation. In *Proceedings of ArabicNLP 2023*, pages 702–707.

Mohamed Basem, Islam Oshallah, Baraa Hikal, Ali Hamdi, and Ammar Mohamed. 2024. Optimized quran passage retrieval using an expanded qa dataset and fine-tuned language models. In *The International Conference of Advanced Computing and Informatics*, pages 244–254. Springer.

Mohammed Alaa Elkomy and Amany Sarhan. 2024. Tce at qur'an qa 2023 shared task: Low resource enhanced transformer-based ensemble approach for qur'anic qa. *arXiv preprint arXiv:2401.13060*.

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 *Proceedinsg 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.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023, Singapore (Hybrid), December 7, 2023*, pages 690–701. Association for Computational Linguistics.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Abdulrezzak Zekiye and Fadi Amroush. 2023. Al-jawaab at qur'an qa 2023 shared task: Exploring embeddings and gpt models for passage retrieval and reading comprehension. In *Proceedings of ArabicNLP 2023*, pages 743–747.

A Evaluating LLM Embeddings on QPC

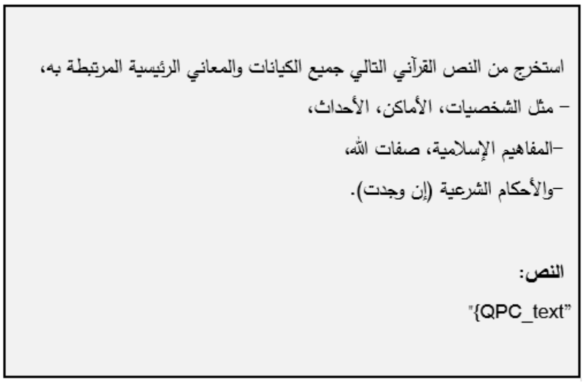
As shown in Table 5, the "Muffakir embeddings" [4]—trained on culturally and religiously aligned Arabic corpora—demonstrated strong performance, outperforming general-purpose multilingual models such as "gte-multilingual-base" [5] and "multilingual-e5-large" [6] models. This suggests that domain-adapted embeddings are more effective at capturing Qur'anic semantics. Although Muffakir is smaller in scale than OpenAI's model, its competitive results underscore the advantages of domain relevance. Meanwhile, the superior performance of text-embedding-3-large is likely due to a combination of advanced model architecture, large-scale multilingual training, and task-specific optimization for retrieval.

Model	MAP@10	MAP@5
gte-multilingual-base	0.1542	0.1429
multilingual-e5-large	0.1814	0.1678
Muffakir Embedding	0.1994	0.1867
text-embedding-3-large	**0.2761**	**0.2553**

Table 5: Performance of different retrieval methods using semantic search approaches on both the train and dev sets.

B Prompt Engineering for Factual Information Extraction from Islamic Texts

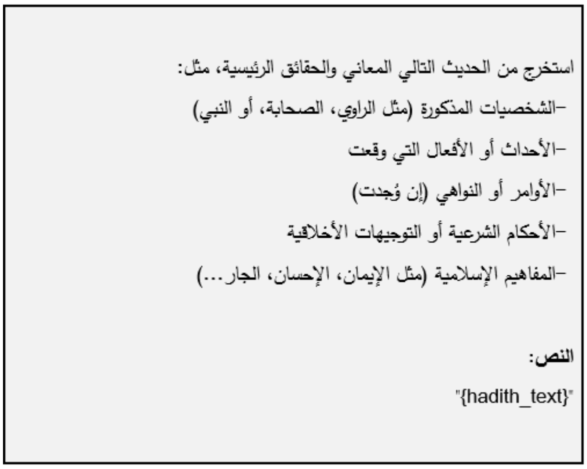
Figures 1 and 2 illustrate examples of prompt designs aimed at extracting factual information from Quranic verses and Hadith texts, respectively.



استخرج من النص القرآني التالي جميع الكيانات والمعاني الرئيسية المرتبطة به،
– مثل الشخصيات، الأماكن، الأحداث،
–المفاهيم الإسلامية، صفات الله،
–والأحكام الشرعية (إن وجدت).

النص:
"{QPC_text"

Figure 1: Example of a prompt designed to extract factual information from Quran verses.



استخرج من الحديث التالي المعاني والحقائق الرئيسية، مثل:
–الشخصيات المذكورة (مثل الراوي، الصحابة، أو النبي)
–الأحداث أو الأفعال التي وقعت
–الأوامر أو النواهي (إن وُجدت)
–الأحكام الشرعية أو التوجيهات الأخلاقية
–المفاهيم الإسلامية (مثل الإيمان، الإحسان، الجار...)

النص:
"{hadith_text}"

Figure 2: Example of a prompt designed to extract factual information from Hadith.

These prompts are specifically crafted to guide LLMs in identifying key Islamic concepts and legal rulings embedded within each Quranic passage or Hadith.

[4] https://huggingface.co/mohamed2811/Muffakir_Embedding

[5] https://huggingface.co/Alibaba-NLP/gte-multilingual-base

[6] https://huggingface.co/intfloat/multilingual-e5-large

Isnad AI at IslamicEval 2025: A Rule-Based System for Identifying Islamic Citation in LLM Outputs

Fatimah Emad Eldin
Department of Computer and Information Sciences
Faculty of Graduate Studies for Statistical Research, Cairo University
12422024441586@pg.cu.edu.eg

Abstract

This paper presents the Isnad AI system developed for the IslamicEval 2025 Shared Task 1A, which focuses on identifying character-level spans of Quranic verses (Ayahs) and Prophetic sayings (Hadiths) within Large Language Model (LLM) outputs. This task is formulated as a token classification problem using a fine-tuned AraBERTv2 model. The primary contribution is a novel rule-based data preprocessing and augmentation pipeline, through which a large-scale, high-quality training corpus is systematically generated from raw religious texts. Through comprehensive ablation studies, it is demonstrated that the controlled synthetic data generation approach significantly outperforms traditional database lookup methods and basic fine-tuning approaches. The system achieved an F1 score of 66.97% in the official test set, demonstrating the effectiveness of principled synthetic data generation for specialized religious text verification tasks. To support reproducibility and future research in Islamic citation detection, all code, generated datasets, and experimental resources are made publicly available on GitHub and Hugging Face.

1 Introduction

The proliferation of Large Language Models (LLMs) has created an urgent need for robust mechanisms to verify factual accuracy (?Li et al., 2024), particularly in specialized domains like Islamic studies (Nagoudi et al., 2022; Antoun et al., 2021). The IslamicEval 2025 Shared Task 1A addresses this by requiring systems to detect precise character-level spans of religious citations within Arabic LLM responses (Mubarak et al., 2025), representing a foundational step for subsequent fact-checking systems. The submitted system employs a token classification method using a fine-tuned AraBERTv2 model (Antoun et al., 2020).

Given the absence of large, manually annotated corpora for this task. A rule-based process was developed to programmatically generate clean, contextualized training data, embedding authentic religious texts within varied templates to simulate LLM citation patterns.

Ablation studies revealed that this rule-based data generation methodology outperforms database lookup, and basic fine-tuning. To foster reproducibility and support future research in Islamic religious citation, all experimental code, dataset, and the final fine-tuned model are publicly available on GitHub[1] and Hugging Face[2].

2 Background

This work addresses the IslamicEval 2025 Shared Task 1A, which requires identifying character-level spans of Quranic verses (Ayahs) and Prophetic sayings (Hadiths) within LLM-generated Arabic text (Mubarak et al., 2025). For a given text containing citations, the system must identify the exact start and end character indices. The required submission format is detailed in the Appendix B.3 in Table 4. This task is structured as a token classification problem using the standard BIO schema to label the boundaries of religious citations (Ramshaw and Marcus, 1995; Devlin et al., 2019).

2.1 Related Work

This work is situated within the broader field of adapting language models for specialized religious domains (Nagoudi et al., 2022). While there has been progress in this area, this verification task presents a unique challenge because existing resources are not suitable for precise, character-level span detection. Foundational datasets like the Quranic Arabic Corpus (Dukes and Habash, 2010) provide deep morphological analysis, and there are

[1] https://github.com/astral-fate/IslamicEval

[2] https://huggingface.co/collections/FatimahEmadEldin/isnad-ai-at-islamiceval-68a64677910651f034b9cdf4

Proceedings of The Third Arabic Natural Language Processing Conference, pages 540–559
November 8-9, 2025 ©2025 Association for Computational Linguistics

Dataset Split	**Unique Texts**	**Ayah Examples**	**Hadith Examples**	**Total Generated**
Training Set	30,548	20,622	72,477	93,099
Validation Set	13,354	20,313	20,313	40,626
TOTAL	**43,902**	**40,935**	**92,790**	**133,725**

(a) Final generated splits with class breakdown.

Corpus	**Original Count**
Quranic Verses (Ayahs)	6,236
Hadith Narrations	34,662
Total Unique Texts	**40,898**

(b) Original source data.

Corpus	**Preprocessed Count**
Total Unique Ayahs	13,456
Total Unique Hadiths	30,446
Total Unique Texts	**43,902**

(c) After preprocessing.

Table 1: Corrected dataset statistics at each stage. Table (a) shows the final splits and total generated examples based on the actual output files. Table (b) shows the initial counts from source files. Table (c) shows the total number of unique texts available for splitting after all processing and augmentation.

models fine-tuned for Islamic question-answering (Ellbendis, 2024; Justdeen, 2024).

However, these resources were not designed for the specific purpose of identifying exact citation boundaries within a larger text.

This creates a significant data scarcity problem for this particular task. To address this gap, the primary contribution of this work is a novel rule-based data generation pipeline. This approach was developed to create a suitable, large-scale training corpus, directly overcoming the lack of annotated data for this specialized verification task (Hedderich et al., 2021).

3 System Overview

3.1 Core Model

The foundation of the system is a fine-tuned implementation of AraBERTv2 (Antoun et al., 2020), a powerful transformer-based model pre-trained on a large corpus of Arabic text. For this task, the model was adapted for token classification and fine-tuned to predict labels according to the standard BIO schema: B-Ayah, I-Ayah, B-Hadith, I-Hadith, or O (Outside). Through this approach, the system can effectively identify the precise boundaries of religious citations at a granular level within LLM-generated text. The model was trained exclusively on the synthetically generated dataset, which is detailed in section 4.

3.2 Training Data Generation

The central methodological contribution is the programmatic generation of a large-scale training corpus. This approach was developed to overcome the lack of manually annotated data by creating high-quality, contextualized examples to simulate how they appear as in-context citations within LLM outputs. The process is detailed in section 4.3.

4 Data and Preprocessing Pipeline

The entire training and validation dataset was synthetically generated from raw Islamic texts using a multi-stage pipeline designed to create diverse and realistic training examples.

4.1 Data Sources

Two foundational datasets of sacred Islamic texts were utilized for this paper. These datasets, provided in a pre-processed format by the task organizers, consist of the following:

- **The Holy Quran** (KFG, 2025): The complete text of the Holy Quran, presented in a JSON file where each entry corresponds to a specific verse (*ayah*) [3].
- **The Hadith**: A collection of prophetic traditions (narrations) from the Six Major Books of Hadith, provided in a JSON file [4].

For model fine-tuning, only Hadith entries containing a non-empty 'Matn' (the core narrative text of the prophetic tradition) were used. The initial

[3] https://github.com/qcri/IslamicEval-2025-Subtask-1/blob/main/Quran/quranic_verses.json

[4] https://github.com/qcri/IslamicEval-2025-Subtask-1/blob/main/Hadith/six_hadith_books.json

distribution of these datasets is summarized in Table 1b.

4.2 Data Preprocessing and Augmentation Pipeline

The preprocessing pipeline systematically transforms raw Islamic texts into a comprehensive training corpus through five interconnected stages (detailed methodology in Appendix D). The process begins with systematic text segmentation and Arabic script normalization, followed by template-based contextual generation that embeds authentic religious texts within realistic citation patterns. The complete pipeline workflow is illustrated in Figure 1, Figure 2 and Figure 3.

1. **Text Splitting**: Quranic verses were analyzed using the AraBERTv2 tokenizer. Any verse exceeding a 25-token length was split into two smaller, more manageable segments. This process increased the total number of Ayah from 6,236 to 6,910.

2. **Normalization and Augmentation**: To improve the model's robustness against variations in Arabic script, a data augmentation technique was applied. For every Ayah (both original and segmented), a duplicate version was created with all diacritics (*Tashkeel*) removed.

3. **Template-Based Generation**: The core of the pipeline involves embedding the processed religious texts into contextual templates. A set of common prefixes and suffixes were manually curated for both Ayahs and Hadiths based on a qualitative analysis of common citation patterns in contemporary Arabic writing. The lists in Table 13 provide the comprehensive examples of suffix and prefix of the rule-based templates.

The data distribution after these preprocessing and augmentation is shown in Table 1c.

4.3 Dataset Splits

The synthetic data generation pipeline produced a corpus from 43,902 unique religious texts. This corpus was split into the internal training and validation sets to fine-tune the AraBERTv2 model. A 70/30 split was employed, allocating 70% of the unique source texts for the training set and 30% for the internal validation set. The template-based generation process was then applied to these partitioned texts, resulting in the final example counts shown in Table 1a. For final evaluation, the official datasets provided by the shared task organizers was used. The model's performance on the development set (referred to as ”Dev F1” in Table 2) was measured against the organizers' manually annotated ‘dev SubtaskA‘ files, containing 210 records. The final competition score (referred to as ”Test F1”) was evaluated on the official blind test set of 190 records. The internal validation set was used exclusively for hyperparameter tuning and to prevent overfitting during the fine-tuning phase.

Methodology	**Dev F1**	**Test F1**
Rule-Based Model	65.08%	**66.97%**
Ablation Baselines:		
Database Lookup	52.00%	34.80%
Basic Fine-Tuning	33.00%	44.70%

Table 2: Comprehensive results across development and test sets compared to ablation baselines.

5 Experimental Setup

5.1 Evaluation Metric

The official evaluation metric for the task is the Macro-Averaged F1 Score computed at the character level (Mubarak et al., 2025). Unlike span-based evaluation, this metric treats each character of the response string as an independent classification unit, assigning it one of three labels: Ayah, Hadith, or Neither. The F1 score is then computed as the harmonic mean of Precision (P) and Recall (R). The macro-averaged F1 score computes the F1 score for each class independently and then averages them, giving equal weight to each class regardless of its frequency. This character-level evaluation ensures the system is assessed on its ability to precisely identify the boundaries of religious texts at the finest granularity, making it more stringent than span-based metrics (Tjong Kim Sang and De Meulder, 2003).

6 Results

6.1 Ablation Study Analysis

Comprehensive ablation studies were conducted to evaluate the proposed rule-based synthetic data generation approach against two baseline methodologies: database lookup and basic fine-tuning without synthetic augmentation. The experimental results demonstrate substantial superiority of the rule-based model across both evaluation sets. As presented in Table 2, the rule-based approach achieved macro F1 scores of 65.08% on the development set and 66.97% on the official test set. The baseline models performed significantly worse on the development set, with the database lookup method achieving an F1 score of 52% and the basic fine-tuning approach achieving 33%. While both baselines showed limitations, the results validate the effectiveness of principled synthetic data generation, demonstrating a performance improvement of 22.27% over basic fine-tuning on the official test set.

7 Error Analysis

The error analysis was conducted on the development set, detailed in Appendix F, as the ground truth for the final blind test set was not provided by the shared task organizers.

7.1 Impact of Class Imbalance

A significant class imbalance exists, with the 'Neither' class comprising 67.8% of characters, while 'Ayah' and 'Hadith' account for only 20.2% and 12.0%, respectively (see Appendix C). This class imbalance is reflected in the F1-scores: 0.90 for the majority 'Neither' class, 0.67 for 'Ayah', and a significantly lower 0.39 for the 'Hadith' class. The primary weakness is identifying Hadith, a challenge compounded by their narrative style and significant textual variance across different Hadith books, making them harder to distinguish, in comparison to Quranic verses.

7.2 Span-Level Error Patterns

A span-level analysis reveals the model produced more False Negatives (101 missed spans) than True Positives (78 correct spans). Missed spans were comparable in length to correctly identified ones, suggesting the model tends to miss entire citations. Conversely, False Positives were predominantly short fragments, indicating a tendency to misclassify small, unrelated phrases.

8 Discussion

The experimental results highlight the critical role of data quality in training models for specialized verification tasks. Several approaches were evaluated, including a database lookup method and basic fine-tuning. A generative synthetic data approach using AraGPT2 (Antoun et al., 2021) was evaluated; however, the generative synthetic data proved inappropriate. Using the prompt templates shown in Table 14, the model produced significant noise. As detailed in Appendix G and exemplified in Table 15, the outputs included nonsensical fragments and contextual hallucinations, creating misleading training data. These results validate that the structured, rule-based approach to synthetic data generation was the most effective strategy for this task. The system's primary challenge remained in Hadith identification, where performance was hindered by significant textual variation in narrations across the six major books of Hadith. This high degree of narrative variation, unlike the uniformity of Quranic verses, poses a significant modeling challenge.

9 Conclusion

This paper presented the Isnad AI system for identifying religious citations in LLM outputs using fine-tuned AraBERTv2 with a novel rule-based synthetic data generation pipeline. The system achieved 66.97% F1 on the test set, significantly outperforming database lookup (34.80%) and basic fine-tuning (44.70%), validating the effectiveness of principled synthetic data generation for specialized verification tasks. The primary limitation was Hadith identification (F1: 0.39 vs. Quranic verses: 0.67), attributed to the textual variation of the Matn across different narrators. Future work should confine training to a single Hadith book, such as Sahih al-Bukhari (al Bukhari, 1871), explore class-balanced sampling, and develop techniques for detecting corrupted or paraphrased citations. The latter could be achieved by enhancing the lookup baseline with fuzzy matching algorithms or by augmenting the training data with synthetically generated textual variations to improve the deep learning model's robustness.

Acknowledgments

The organizers of IslamicEval 2025 Shared Task are gratefully acknowledged for their efforts in creating this important benchmark for Arabic religious text processing.

References

2025. *Al-Qur'an al-Karim (Mushaf al-Madinah an-Nabawiyyah)*. King Fahd Complex for the Printing of the Holy Qur'an, Medina, Saudi Arabia. Arabic text based on the Uthmanic (script), in the narration of Hafs from Asim. Corresponds to Hijri year 1446-1447 AH.

Muhammad Ismail al Bukhari. 1871. *Sahih al-Bukhari*. King Fahd National Library - Riyadh.

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kais Dukes and Nizar Habash. 2010. Morphological annotation of Quranic Arabic. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Ellbendis. 2024. Qwen3-4b-quran-lora-fine-tuned. https://huggingface.co/Ellbendls/Qwen3-4b-Quran-LoRA-Fine-Tuned. Accessed: 2025-08-16.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Justdeen. 2024. Quranplus. https://huggingface.co/justdeen/QuranPlus. Accessed: 2025-08-16.

Siheng Li, Cheng Yang, Taiqiang Wu, Chufan Shi, Yuji Zhang, Xinyu Zhu, Zesen Cheng, Deng Cai, Mo Yu, Lemao Liu, Jie Zhou, Yujiu Yang, Ngai Wong, Xixin Wu, and Wai Lam. 2024. A survey on the honesty of large language models. *arXiv preprint arXiv:2409.18786*.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. AraT5: Text-to-text transformers for Arabic language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 628–647, Dublin, Ireland. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

A Experimental Configuration

Table 3 provides the hyperparameter settings used for fine-tuning the AraBERTv2 model (Antoun et al., 2020).

Parameter	Value
Model	aubmindlab/bert-base-arabertv2
Max Epochs	10 (with early stopping)
Learning Rate	2×10^{-5}
Batch Size (per device)	4
Gradient Accumulation Steps	4
Effective Batch Size	16
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	500
Mixed Precision	fp16 enabled
Max Sequence Length	512 tokens
Early Stopping Patience	3 epochs

Table 3: Complete hyperparameter configuration for model training.

Question_ID	Span_Start	Span_End	Span_Type
A-Q001	11	25	Ayah
A-Q001	34	52	Ayah
A-Q001	67	87	Hadith

Table 4: Example submission file format.

B Data Structures and Format Specifications

This appendix provides detailed specifications of the data structures used throughout the system, including source data formats and submission requirements.

B.1 Source Data Structures

Quranic Verses The Quranic data is structured as a JSON array, in which each object corresponds to a single verse.

```
{
  ”surah_id”: 1,
  ”surah_name”: ”الفاتحة”,
  ”ayah_id”: 1,
  ”ayah_text”: ”بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ”
},
```

Hadith The Hadith data was structured as a JSON array, with each object representing a Hadith.

```
{
  ”hadithID”: 5,
  ”BookID”: 1.0,
  ”title”: ”...”,
  ”hadithTxt”: ”...”,
  ”Matn”: ”إِنَّمَا الأَعْمَالُ بِالنِّيَّاتِ...”
},
```

B.2 Test Data Format

The test data is in XML format, with each <Question> block containing the LLM's response.

```
<Question>
  <ID>A-Q001</ID>
  <Model>Model-6</Model>
  <Text>هل يمكن أن يكون
الابتلاء....</Text>
  <Response>
    نعم، يمكن أن يكون الابتلاء....

    </Response>
</Question>
```

B.3 Submission Format

The required submission is a Tab-Separated Values (TSV) file with the columns: Question_ID, Span_Start, Span_End, and Span_Type. An example is shown in Table 4.

C Development Set: Exploratory Data Analysis

To better understand the composition of the dataset used for evaluation, an exploratory data analysis (EDA) was performed on the development set. This set consists of 50 questions and their corresponding responses, containing a total of 210 manually annotated spans of text. The analysis was conducted at the character level to align with the official scoring methodology. The primary finding is a significant class imbalance within the data, as detailed in Table 5a. The 'Neither' class, representing text that is not part of a religious quotation, constitutes over two-thirds of the total characters. The 'Ayah' class is the most represented quotation type, accounting for 20.2% of the characters, while the 'Hadith' class is the least represented at 12.0%. Further analysis of the annotated spans, summarized in Table 5b, reveals additional insights. There are more distinct 'Ayah' spans (118) than 'Hadith' spans (76). A notable characteristic is the high variance in the length of these spans. For both classes, the standard deviation is nearly as large as the mean, and the lengths range from very short fragments to extensive passages of over 600 characters. This indicates that the model must be capable of identifying quotations of highly variable lengths.

D Comprehensive Data Preprocessing Pipeline

This appendix details the rule-based data preprocessing pipeline designed to transform raw Islamic texts into a high-quality training corpus for token classification. The architecture is composed of five sequential stages: (1) Data Acquisition and Validation, (2) Text Preprocessing and Augmentation, (3) Template-Based Data Generation, (4) Dataset Partitioning, and (5) Tokenization with Label Assignment.

D.1 Stage 1: Data Acquisition and Validation

The initial corpus was established from two primary sources provided by the shared task organizers: the Quranic corpus, containing 6,236 verses (Ayahs), and a Hadith collection of 34,662 prophetic narrations. During acquisition, textual content was extracted from designated fields within the source JSON files: 'ayah text' for the Quran and 'Matn' (the core narrative) for the Hadith. To ensure corpus integrity and manage com-

Class	Count	Pct.
Neither	44,273	67.8%
Ayah	13,173	20.2%
Hadith	7,864	12.0%
Total	**65,310**	**100.0%**

(a) Character-level class distribution.

Class	Spans	Mean	Std.	Min	Max
Ayah	118	111.6	105.1	6	678
Hadith	76	103.5	93.0	8	690

(b) Descriptive statistics for annotated spans.

Table 5: Summary of the development set's class distribution by character count (a) and annotated span statistics (b).

putational resources, two validation measures were implemented:

- **Length Threshold:** A maximum text length of 1,500 characters was imposed to prevent memory overflow, while retaining the vast majority of authentic texts.
- **Encoding Validation:** All texts were validated for proper UTF-8 encoding to ensure correct handling of the Arabic script.

D.2 Stage 2: Text Preprocessing and Augmentation

This stage addresses linguistic and tokenizer-specific challenges through text segmentation and script normalization.

D.2.1 Text Segmentation

To accommodate the processing limitations of the AraBERTv2 tokenizer, texts exceeding a 25-token threshold were systematically segmented. The segmentation algorithm identifies the approximate midpoint of a text and performs a backward search for the nearest word boundary (whitespace). This content-aware strategy prevents splitting words, thus preserving semantic coherence. This process expanded the initial 6,236 Quranic verses into 6,910 processable text segments. While some Quranic verses remained long even after bisection, the split was limited to two parts to minimize the risk of excessive fragmentation and loss of contextual meaning; multi-part splitting strategies are reserved for future work.

D.2.2 Arabic Script Normalization

To enhance model robustness against script variations, a data augmentation strategy involving diacritic removal was applied. For each Ayah (original and segmented), a normalized variant was generated by removing all diacritical marks (Tashkeel) and the Tatweel character, which correspond to the Unicode range [\u064B-\u0652\u0640]. This technique effectively doubled the unique Ayah corpus to 13,456, ensuring the model can recognize verses regardless of their vocalization.

D.3 Stage 3: Template-Based Contextual Generation

This stage is the core of the synthetic data generation pipeline, designed to programmatically create a large-scale, high-quality training corpus. The primary objective is to embed the clean, preprocessed religious texts from the previous stage into varied contextual templates, thereby simulating the patterns commonly observed when Large Language Models (LLMs) cite religious sources. The generation process is algorithmic and designed to produce multiple unique examples from a single source text, significantly augmenting the dataset. For each source text (either a Quranic Ayah or a Hadith), the system executes the following steps, as illustrated in the workflow diagram in Figure 2 3:

1. **Template Component Selection**: Based on the text's label (Ayah or Hadith), the system randomly selects a corresponding prefix and suffix from a manually curated list. These lists, detailed in the paper's Table 13, contain common introductory and concluding phrases used in contemporary Arabic writing to cite religious texts.
2. **Contextual Enrichment**: To enhance the realism of the generated examples, a neutral or transitional sentence is added with a 30% probability. This sentence is randomly selected from a predefined list and is inserted either before or after the religious text to break simplistic patterns and better mimic the flow of natural language.
3. **Concatenation and Normalization**: The selected components are combined in one of the following structures:

- suffix + source_text + prefix
- + source_text + neutral_context + prefix suffix
- + neutral_context + source_text + prefix suffix

The resulting string is then normalized to ensure consistent spacing.

4. **Span Detection and Labeling**: The exact start and end character indices of the original source_text are programmatically located within the final concatenated string (full_text). This step is critical for creating the precise character-level annotations required for training.

This automated process was applied to the partitioned source texts, ultimately generating **93,099** examples for the training set and **40,626** for the validation set. The final output is a structured dataset where each entry contains the original text, the newly generated contextual sentence, and the precise citation boundaries. An example of the final generated data structure is shown in Table 6.

D.4 Stage 4: Dataset Partitioning

A systematic partitioning strategy was applied at the source text level to create distinct training and validation sets. The corpus of unique source texts was split using a 70-30 ratio, allocating 30,548 texts for training and 13,354 for internal validation. This split was performed using stratified sampling to preserve the original distribution of Ayah and Hadith texts in both partitions. A fixed random seed (42) was used to ensure the reproducibility of the splits.

D.5 Stage 5: Tokenization and Label Assignment

The final stage converts the generated examples into a format suitable for model training.

- **BIO Schema:** A five-class BIO (Beginning-Inside-Outside) tagging schema was employed: O (Outside), B-Ayah, I-Ayah, B-Hadith, and I-Hadith. This schema allows the model to learn the precise boundaries of each citation type.

- **Tokenization and Alignment:** Each example was tokenized using the AraBERTv2 tokenizer with a maximum sequence length of 512 tokens. The character-level span indices were mapped to their corresponding token indices. The first token of a span was assigned the 'B' label, subsequent tokens within the span received the 'I' label, and all other tokens were labeled 'O'. To handle subword tokenization, continuation tokens within a word were assigned an ignore index of -100, ensuring that the loss function only considers the primary token of each word.

D.5.1 Validation Framework

A multi-tiered validation approach was used for comprehensive performance assessment.

- **Internal Validation Set:** Created from the 30% partition of the original source texts. This set, containing approximately 40,626 synthetically generated examples, was used exclusively for hyperparameter optimization and model selection during development. To test generalization, the templates used to generate these examples differed from those used for the training set.

- **Official Development Set:** A set of 210 manually annotated examples provided by the task organizers. This set was used to evaluate the model's ability to generalize from synthetic data to authentic LLM-generated content.

- **Official Test Set:** A blind set of 190 examples used for the final competitive evaluation. Performance on this set determined the final reported scores.

D.5.2 Quality Control

Several measures were implemented to ensure the integrity of the generated dataset:

- **Failure Tracking:** Generation failures, such as span detection errors, were tracked, and only successfully generated examples were included in the final corpus.

- **Data Validation:** Routine checks were performed to verify character encodings, label consistency, and the integrity of tokenizer outputs.

- **Statistical Monitoring:** Statistics on the class distribution and the ratio of source texts to generated examples were monitored for transparency.

original_text	full_text	prefix	suffix	char_start	char_end	label_type	target_span	variation_number	dataset_split
وأمددناهم بفاكهة ولحم مما يشتهون	ومن آيات الله: وفي هذا هداية للمؤمنين. "وأمددناهم بفاكهة ولحم مما يشتهون" آية كريمة	ومن آيات الله:	آية كريمة	40	72	Ayah	وأمددناهم بفاكهة ولحم مما يشتهون	1	training
فويل يومئذ للمكذبين	ومن آيات الله: "فويل يومئذ للمكذبين" ولذلك عبرة للمعتبرين	ومن آيات الله:	ولذلك عبرة للمعتبرين	16	35	Ayah	فويل يومئذ للمكذبين	2	training

Table 6: Example of Final Generated Data Structure

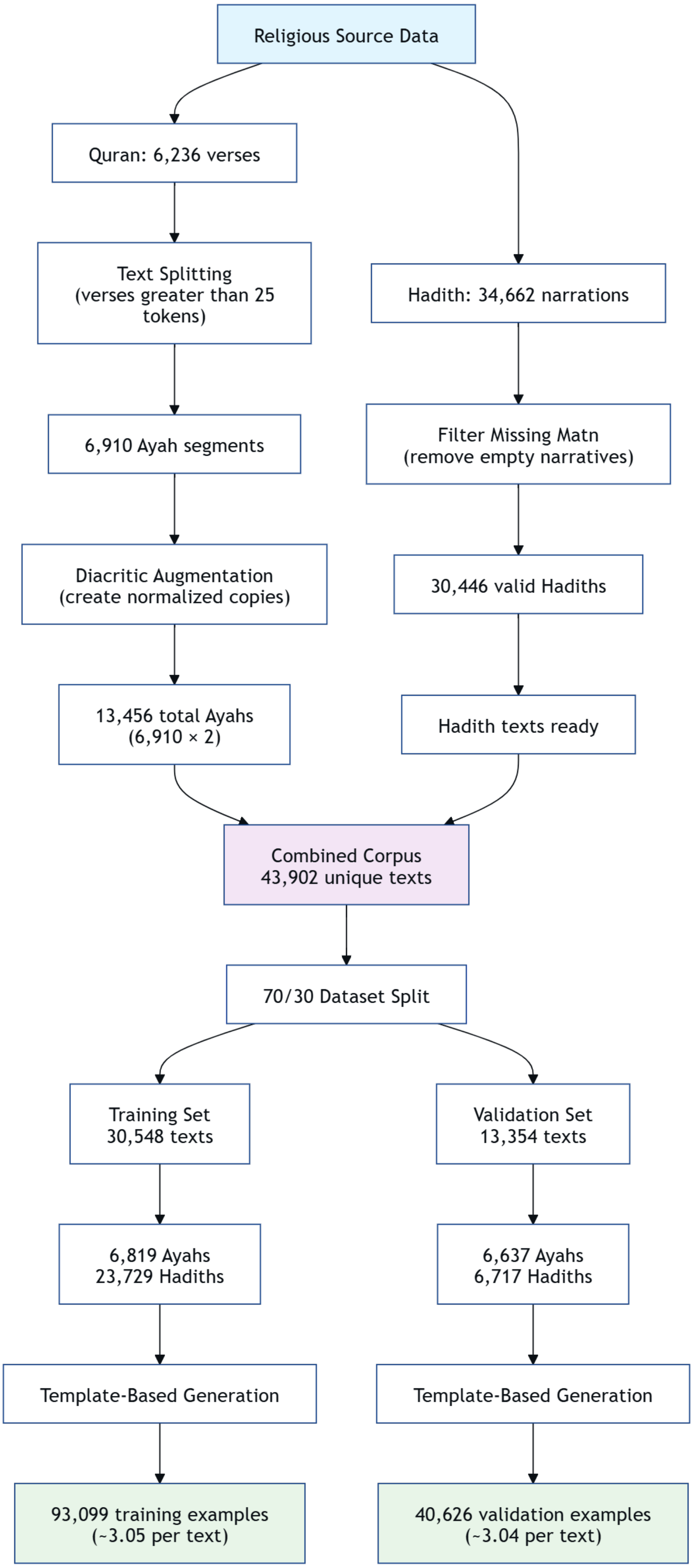


Figure 1: Data preprocessing pipeline transforming 40,898 raw Islamic texts into 133,725 training and validation examples through filtering, augmentation, and template-based generation.

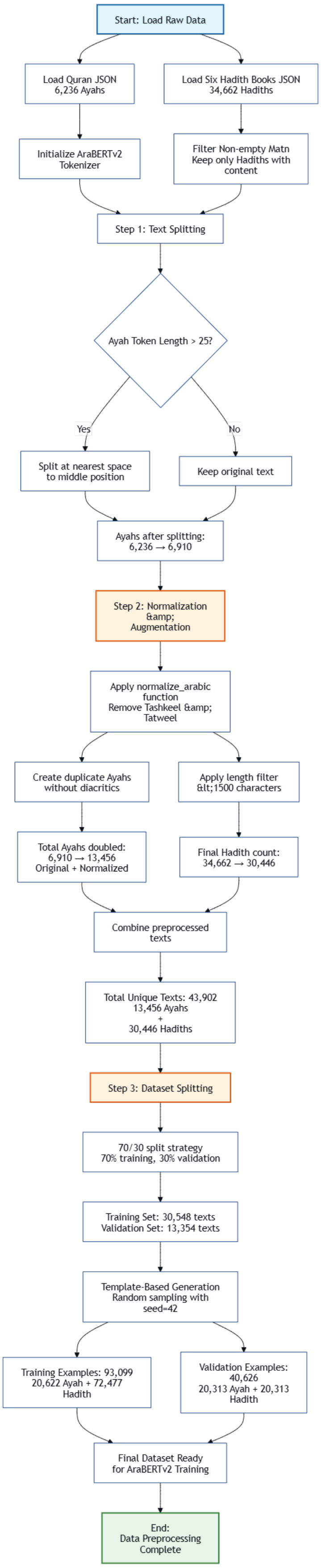


Figure 2: A high-level diagram of the data prepossessing pipeline

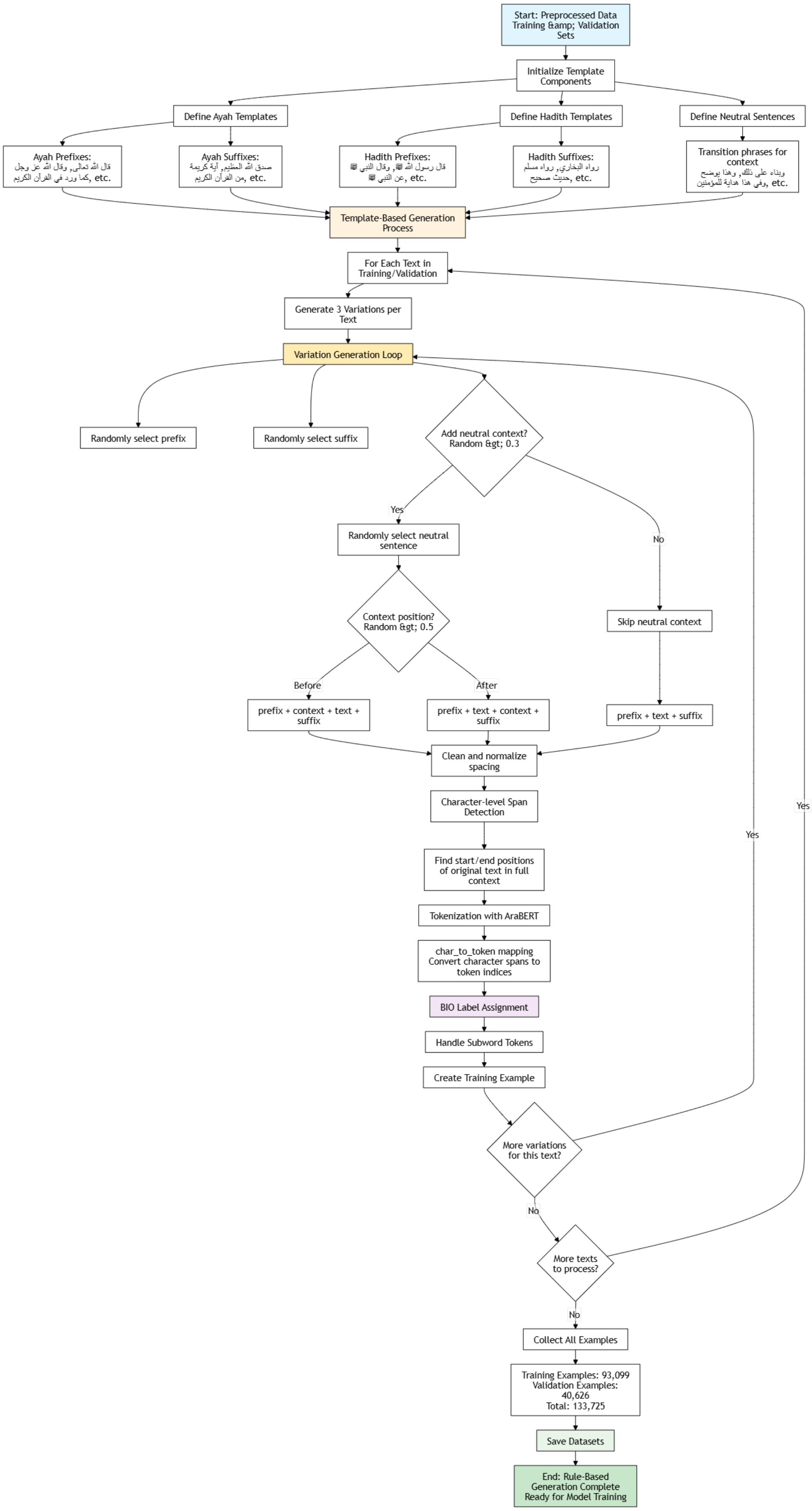
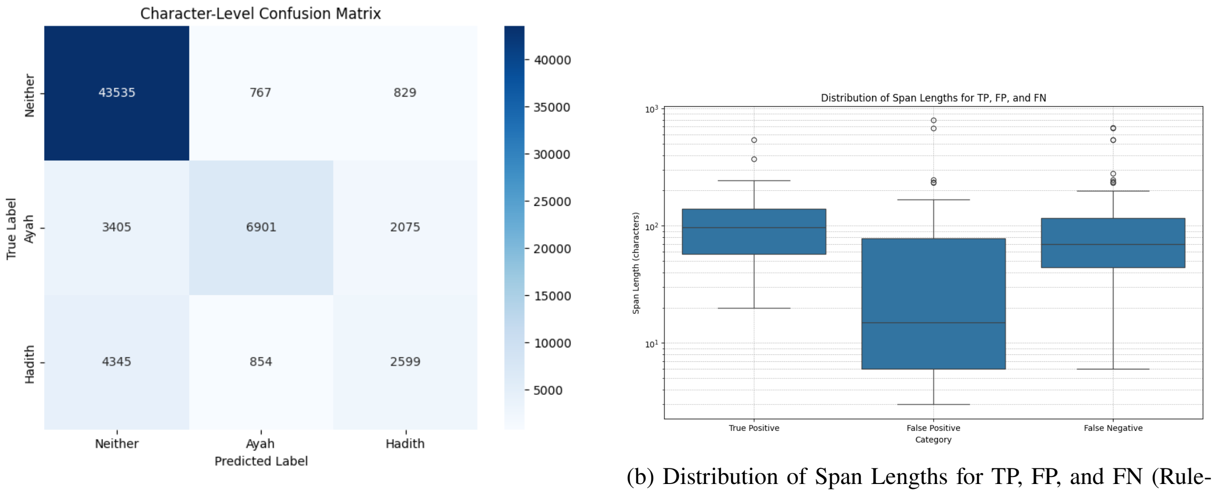

Start: Preprocessed Data Training &amp; Validation Sets
Initialize Template Components
Define Ayah Templates
Define Hadith Templates
Define Neutral Sentences
Ayah Prefixes: قال الله تعالى, وقال الله عز وجل كما ورد في القرآن الكريم, etc.
Ayah Suffixes: صدق الله العظيم, آية كريمة من القرآن الكريم, etc.
Hadith Prefixes: قال رسول الله ﷺ, وقال النبي ﷺ عن النبي ﷺ, etc.
Hadith Suffixes: رواه البخاري, رواه مسلم حديث صحيح, etc.
Transition phrases for context وبناء على ذلك, وهذا يوضح وفي هذا هداية للمؤمنين, etc.
Template-Based Generation Process
For Each Text in Training/Validation
Generate 3 Variations per Text
Variation Generation Loop
Randomly select prefix
Randomly select suffix
Add neutral context? Random &gt; 0.3
Yes
Randomly select neutral sentence
No
Skip neutral context
Context position? Random &gt; 0.5
Before
After
prefix + context + text + suffix
prefix + text + context + suffix
prefix + text + suffix
Clean and normalize spacing
Character-level Span Detection
Find start/end positions of original text in full context
Tokenization with AraBERT
char_to_token mapping Convert character spans to token indices
BIO Label Assignment
Handle Subword Tokens
Create Training Example
Yes
Yes
More variations for this text?
No
More texts to process?
No
Collect All Examples
Training Examples: 93,099 Validation Examples: 40,626 Total: 133,725
Save Datasets
End: Rule-Based Generation Complete Ready for Model Training

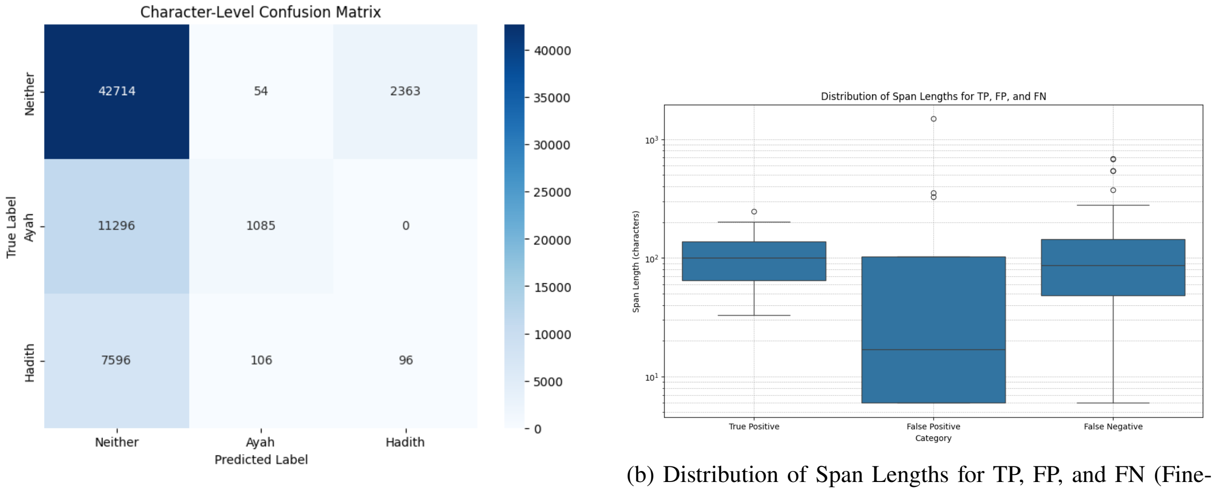
Figure 3: A high-level diagram of the rule-based data generation process.



(a) Character-Level Confusion Matrix (Rule-Based Model)

(b) Distribution of Span Lengths for TP, FP, and FN (Rule-Based Model)

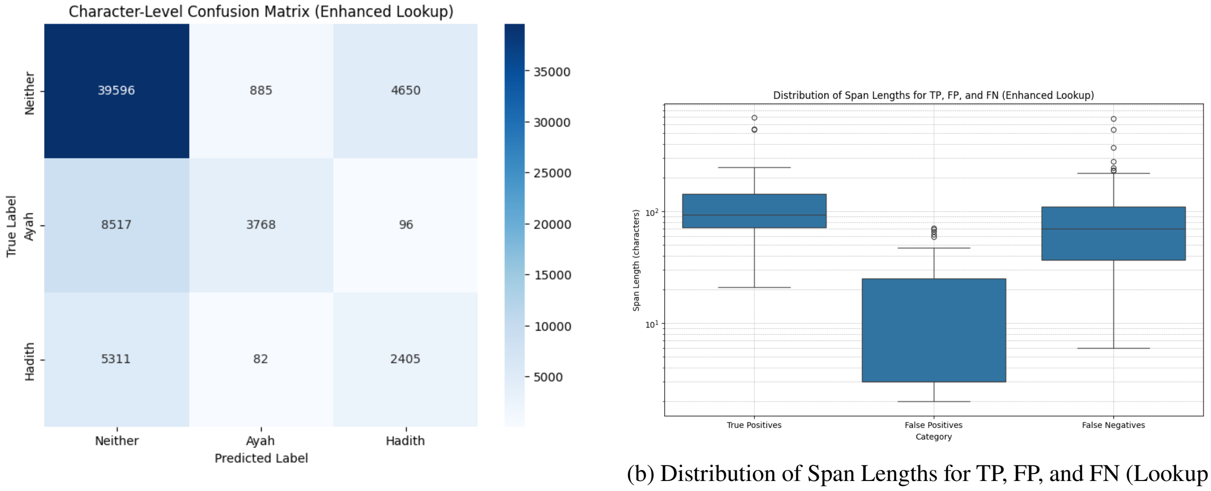
Figure 4: Span-Level Error Logging for the Rule-Based Model.



(a) Character-Level Confusion Matrix (Fine-tuned Model)

(b) Distribution of Span Lengths for TP, FP, and FN (Fine-Tuned Model)

Figure 5: Performance by Span Length for the Basic Fine-Tuning Model.



(a) Character-Level Confusion Matrix (Lookup Method)

(b) Distribution of Span Lengths for TP, FP, and FN (Lookup Method)

Figure 6: Performance by Span Length for the Lookup Method.

E Database Lookup Methodology

This appendix provides a detailed, step-by-step description of the database lookup method, which was implemented as a key baseline in the ablation study (see Table 2). This method relies on direct string matching against an enhanced knowledge base, serving as a non-neural benchmark to evaluate the performance of the fine-tuned model. The entire process can be broken down into two main stages: (1) Knowledge Base Enhancement and (2) The Span Detection Algorithm.

E.1 Stage 1: Knowledge Base Construction and Enhancement

The effectiveness of a lookup-based approach is highly dependent on the comprehensiveness of its knowledge base. To maximize the chances of finding a match, the raw source texts were significantly augmented through a multi-step enhancement pipeline.

E.1.1 Initial Data Loading

The process begins by loading the complete set of Quranic verses and Hadith narrations from the source JSON files provided by the task organizers (quran.json and six_hadith_books.json). The core textual content is extracted from the ayah_text field for Quranic verses and the Matn field for Hadiths. These texts form the initial, unprocessed knowledge base.

E.1.2 Arabic Script Normalization

To handle variations in Arabic script and vocalization, a normalization function was applied to every text in the knowledge base. This function removes all Arabic diacritics (*Tashkeel*) and the Tatweel character by targeting the Unicode range [\u064B-\u0652\u0640]. This step is crucial because LLM outputs may not include the same diacritics as the canonical source texts, and this normalization makes the matching process robust against such differences.

E.1.3 Text Segmentation for Partial Matching

LLMs often cite partial verses or fragmented Hadiths. To account for this, a text segmentation strategy was implemented. Any text (both original and normalized) is split into smaller, overlapping segments. The algorithm generates segments ranging from a minimum of 5 words to a maximum of 15 words, with a step size of 3 words. This process creates a large set of smaller text chunks. For example, a 20-word Hadith would be broken down into multiple 5-word, 6-word, ..., up to 15-word segments. This significantly increases the likelihood of detecting a partial citation.

E.1.4 Final Knowledge Base Aggregation

The final, enhanced knowledge base is an aggregation of multiple text variations for each original Ayah and Hadith. For each source text, the knowledge base contains:

1. The original, unaltered text.

2. The normalized (diacritic-free) version of the text.

3. All overlapping segments generated from the original text.

4. All overlapping segments generated from the normalized text.

This augmentation process results in a massive increase in the number of potential strings to search for, thereby improving the recall of the lookup method.

E.2 Stage 2: Span Detection Algorithm

With the enhanced knowledge base constructed, the span detection algorithm processes each LLM response to identify matching text.

E.2.1 Prioritization of Longer Matches

To ensure the quality of the matches, all entries in the enhanced Ayah and Hadith knowledge bases are sorted by string length in descending order. The detection algorithm iterates through these sorted lists, meaning it always attempts to match the longest possible text segments first. This is a critical step that prevents a short, partial match (e.g., a 5-word segment) from being identified if it is already part of a larger, more complete match (e.g., the full 30-word Ayah).

E.2.2 Iterative String Matching

For each LLM response, the algorithm iterates through every entry in the sorted knowledge bases (first Ayahs, then Hadiths). It uses a standard substring search to find all occurrences of a given knowledge base entry within the response text.

E.2.3 Overlap Prevention

To avoid redundant or overlapping annotations, the algorithm maintains a character-level boolean array for each response text, which tracks whether a character has already been assigned to a span. When a potential match is found, the algorithm checks this array to see if any character within the candidate span has already been classified. If there is no overlap, the span's start and end indices are recorded, and the corresponding characters in the tracking array are marked as classified. This ensures that once a sequence of text is identified as an Ayah, it cannot also be partially or wholly identified as another Ayah. If, after searching through the entire knowledge base, no spans are found for a given response, a ”No_Spans” entry is recorded for that Question ID, as per the task requirements.

F Appendix: Development Set Error Analysis

This entire error analysis is conducted on the official development set provided by the shared task organizers, which consists of 210 manually annotated records.

F.1 Rule-Based Model Development Results

The rule-based model achieved a Macro F1 of **65%** on the development set. The detailed character-level report is shown in Table 7.

Class	Precision	Recall	F1-Score
Neither	0.85	0.96	0.90
Ayah	0.81	0.56	0.66
Hadith	0.47	0.33	0.39
Accuracy			**0.81**
Macro Avg	**0.71**	**0.62**	**0.65**
Weighted Avg	**0.80**	**0.81**	**0.80**

Table 7: Character-Level Classification Report for the Rule-Based Model.

F.1.1 Further Error Analysis

Table 8 provides descriptive statistics for the lengths of true positive, false positive, and false negative spans.

Category	Count	Mean	Min	Max
True Positives	78	108.59	20	541
False Positives	61	69.62	3	795
False Negatives	101	104.78	6	690

Table 8: Span Length Statistics (Rule-Based Model).

F.2 Basic Fine-Tuning Development Results

The basic fine-tuning model achieved a Macro F1 of **33%** on the development set. The detailed report is shown in Table 9.

Class	Precision	Recall	F1-Score
Neither	0.69	0.95	0.80
Ayah	0.87	0.09	0.16
Hadith	0.04	0.01	0.02
Accuracy			**0.67**
Macro Avg	**0.53**	**0.35**	**0.33**
Weighted Avg	**0.65**	**0.67**	**0.59**

Table 9: Development set classification report for the basic fine-tuning approach.

F.2.1 Further Error Analysis

Table 10 presents the descriptive statistics for span lengths.

Category	Count	Mean	Min	Max
True Positives	12	111.33	33	247
False Positives	13	184.46	6	1488
False Negatives	173	110.21	6	690

Table 10: Span Length Statistics (Basic Fine-Tuning).

F.3 Database Lookup Development Results

The database lookup approach achieved a Macro F1 of **52%** on the development set. The detailed classification report is shown in Table 11.

Class	Precision	Recall	F1-Score
Neither	0.74	0.88	0.80
Ayah	0.80	0.30	0.44
Hadith	0.34	0.31	0.32
Accuracy			**0.70**
Macro Avg	**0.62**	**0.50**	**0.52**
Weighted Avg	**0.70**	**0.70**	**0.68**

Table 11: Development set classification report for the database lookup approach.

The database lookup approach shows a significant class imbalance in its performance. While it achieves high recall for the ”Neither” class (88%), its ability to identify religious texts is limited. For ”Ayah” spans, the model has good precision (80%) but low recall (30%), indicating it is confident when it makes a prediction but misses many actual verses. The performance on ”Hadith” spans is poor across all metrics (F1-score of 32%). This model's tendency to over-predict the ‘Neither‘ class highlights the inherent difficulty of relying solely on exact-match lookups for this task.

F.3.1 Further Error Analysis

Table 12 provides descriptive statistics for span lengths of the lookup method.

Category	Count	Mean	Min	Max
True Positives	57	130.49	21	690
False Positives	467	12.19	2	71
False Negatives	109	93.11	6	677

Table 12: Span Length Statistics (Lookup Method).

G Generative Data Augmentation Ablation Study

As referenced in the discussion, an ablation study was conducted to evaluate the efficacy of using a generative Large Language Model for synthetic data augmentation. This approach, was compared against the primary rule-based methodology to determine its suitability for creating a training corpus. This appendix details the complete methodology, from data preprocessing to the final generation of contextualized examples.

Methodology

The generative approach utilized the aubmindlab/aragpt2-base model, a transformer-based model for Arabic language generation, accessed via the Hugging Face ‘transformers‘ library. The core strategy was to embed authentic religious texts into open-ended prompt templates and have the model generate a plausible continuation, thereby creating a full, contextualized sentence around the original text.

1. Data Preprocessing

Before being used in prompts, the raw source texts underwent several preprocessing steps to increase data diversity and manage sequence length:

- **Text Loading**: The full set of Quranic verses (*Ayahs*) and Prophetic narrations (*Hadiths*) were loaded from their respective source JSON files.
- **Text Splitting**: Quranic verses exceeding a 25-token limit (as determined by the AraBERTv2 tokenizer) were split into two smaller segments. This was done to prevent truncation and ensure the model could process the entire text.
- **Normalization Augmentation**: To make the model robust to script variations, a duplicate version of each Ayah was created with all diacritics (*Tashkeel*) removed. The final pool of texts for generation included originals, split segments, and their normalized counterparts.

G.1 Template Examples for Data Generation

The core of the generative augmentation strategy involved embedding authentic religious texts within specific prompt templates to simulate natural-language citations. As shown in Table 14, these prompts were designed to frame the religious text as evidence or a quotation within a larger sentence. During the generation process, the text placeholder was dynamically replaced with a Quranic verse or Hadith, which was then used to prompt the AraGPT2 model to generate a contextual continuation.

As referenced in Section 8, an ablation study was conducted to evaluate the efficacy of using a generative Large Language Model for synthetic data augmentation. This approach was compared against the primary rule-based methodology to determine its suitability for creating a training corpus for the verification task. This appendix details the methodology, the prompt templates used, and the analysis of its significant limitations.

G.1.1 Generative Process

For each religious text, the following generative process was executed:

1. A prompt template was selected at random from the list above.
2. The religious text was inserted into the template.
3. The complete prompt was passed to the AraGPT2 text-generation pipeline with specific parameters:
    - **max_new_tokens=30**: To generate a short, contextual continuation rather than a long, potentially divergent paragraph.
    - **no_repeat_ngram_size=2**: To prevent the model from getting stuck in repetitive loops and improve the quality of the generated text.
4. The model’s output, a new, longer string containing the original text, was captured as the ‘full text‘.

5. Finally, the character start and end indices of the original ‘span text‘ were located within the newly generated ‘full text‘ to create the final labeled data point. A fallback mechanism was included to use the prompt itself if the generation process failed.

Table 16 provides examples of the final structured data produced by this pipeline.

G.2 Limitations and Analysis of Generated Data

While the objective was to create diverse training examples, the generative methodology proved inappropriate for this verification task.The outputs were frequently plagued by factual inaccuracies, nonsensical statements, and linguistic artifacts, introducing significant noise into the training data.For a verification task in a sensitive domain like Islamic studies, the integrity of the source text and its context is paramount.The generative model’s tendency to ”hallucinate” or produce illogical continuations is a critical failure that undermines the purpose of the training data, as it creates misleading training signals. Table 15 provides representative examples of these failure modes.

Component	Training Examples	Validation Examples
Ayah Prefixes	قال الله تعالى: وقال الله عز وجل: كما ورد في القرآن الكريم: وفي كتاب الله: ومن آيات الله: يقول سبحانه وتعالى: وفي هذا الشأن يقول الله:	وفي القرآن الكريم نجد: ومن آيات الله: وقد أنزل الله: ويقول الحق تبارك وتعالى: وفي الذكر الحكيم: وفي كتاب الله نقرأ: والدليل على ذلك قوله تعالى:
Ayah Suffixes	صدق الله العظيم آية كريمة من القرآن الكريم كلام الله عز وجل من الذكر الحكيم ولذلك عبرة للمعتبرين وهذا بيان للناس	هذا من كلام الله آية عظيمة من القرآن الكريم كلام رب العالمين من الذكر الحكيم آية كريمة (صدق الله العظيم)
Hadith Prefixes	قال رسول الله صلى الله عليه وسلم: وقال النبي صلى الله عليه وسلم: عن النبي صلى الله عليه وسلم: روى أن النبي صلى الله عليه وسلم قال: وفي الحديث الشريف: وعن أبي هريرة رضي الله عنه قال:	وفي السنة النبوية: ومن هدي النبي صلى الله عليه وسلم: وقد علمنا الرسول صلى الله عليه وسلم: وفي الحديث الشريف نجد: كما جاء في الحديث:
Hadith Suffixes	رواه البخاري رواه مسلم حديث صحيح صلى الله عليه وسلم من السنة النبوية (متفق عليه) أو كما قال صلى الله عليه وسلم	من السنة النبوية حديث نبوي شريف من هدي المصطفى صلى الله عليه وسلم (رواه الترمذي)
Neutral & Transition Sentences	وبناء على ذلك، يمكننا أن نستنتج. وهذا يوضح عظمة التشريع. وفي هذا هداية للمؤمنين. إن في ذلك لآيات لقوم يعقلون. وهذا هو القول الراجح.	ولنتأمل معاً وفي هذا السياق وللتوضيح وإليكم المثال وفي هذا الصدد وهذا يبين لنا أهمية الموضوع.

Table 13: Template for rule-based data generation across training and validation sets.

Component	Arabic Examples	English Translation
Prompt Templates	يُستشهد بالنص التالي: {text}، وهذا يدل على أن	The following text is cited: {text}, and this indicates that
	من الأدلة الشرعية على هذه المسألة، النص التالي: {text}، ويُفهم من ذلك أن	Among the legal evidence for this issue is the following text: {text}, and it is understood from this that
	يتناول النص التالي قضية مهمة، وهي: {text}، حيث أن	The following text addresses an important issue, which is: {text}, as
	يمكن الاستفادة من قوله: {text}، في واقعنا المعاصر عن طريق	We can benefit from the saying: }text{, in the contemporary reality by

Table 14: Prompt templates used for the generative data augmentation experiment. The {text} placeholder was replaced with an authentic religious text.

Original Text (Ayah)	Full Generated Output	Analysis of Failure
وَلَتَنَازَعْتُمْ فِي الْأَمْرِ وَلَٰكِنَّ اللَّهَ سَلَّمَ ۗ إِنَّهُ عَلِيمٌ بِذَاتِ الصُّدُورِ	يمكن الاستفادة من قوله: {text}، في واقعنا المعاصر عن **ا صول تائب التائب طريق المذبون ، جازمونا**	**Nonsensical Artifact**: The generated continuation (in bold) is grammatically incorrect and semantically meaningless. It represents the “illogical artifacts” mentioned in the discussion, creating a confusing and useless training example.
ليأكلوا من ثمره وما	يُستشهد بالنص التالي: {text}، **هذا النوع** وهذا يدل على أن **من الفاكهة له فوائد صحية كثيرة ، حيث أنه يحتوي على نسبة عالية من الألياف الغذائية التي تساعد في الوقاية من العديد من الأمراض مثل السرطان وأمراض القلب**	**Contextual Hallumination**: The model correctly identifies the text fragment as relating to food but proceeds to hallucinate a detailed, modern nutritional context (fiber, cancer, heart disease) that is not present in the original Quranic narrative. This creates a factually incorrect and misleading association.

Table 15: Examples of incorrect and nonsensical data produced by the AraGPT2-based generative augmentation method.

Full Generated Text (full_text)	Original Span (span_text)	Start (char_start)	End (char_end)	Label (label_type)
يُستشهد بالنص التالي: اركض برجلك ۖ، وهذا يدل على أن الله سبحانه وتعالى هو الذي يراك ، وهو الذي يعلمك ما لا تعلم من أمور الدنيا والآخرة. وإذا تأملنا في القرآن الكريم نجد أنه	ۖ اركض برجلك	22	34	Ayah
من الأدلة الشرعية على هذه المسألة، النص التالي: اتَّقُوا اللَّهَ وَقُولُوا قَوْلًا سَدِيدًا، ويُفهم من ذلك أن يذكروا ما هو واجب عليهم ، وما هو مستحب لهم ، وأن يستخدموه في طاعة الله تعالى ورسوله صلى الله عليه وسلم.	اتَّقُوا اللَّهَ وَقُولُوا قَوْلًا سَدِيدًا	48	91	Ayah
يتناول النص التالي قضية مهمة، وهي: أَلَّا تَعْبُدُوا إِلَّا اللَّهَ ۖ، حيث أن الله تعالى قد أمر عباده المؤمنين بأن لا يعبدوا إلا إياه ، وأن لا يتخذوا من دونه أولياء ولا شفعاء ، وأن	ۖ أَلَّا تَعْبُدُوا إِلَّا اللَّهَ	34	59	Ayah

Table 16: Examples of the final structured output from the generative data augmentation pipeline using AraGPT2. This table illustrates the format of the generated data, including the full generated text and the identified character spans of the original religious text embedded within it.

MAHED Shared Task: Multimodal Detection of Hope and Hate Emotions in Arabic Content

Wajdi Zaghouani[1], Md. Rafiul Biswas[2], Mabrouka Bessghaier[1], Shimaa Ibrahim[2], Georgios Mikros[2], Abul Hasnat[3], Firoj Alam[4],
[1]Northwestern University in Qatar, [2]Hamad Bin Khalifa University
[3]APAVI.AI, France, [4]Qatar Computing Research Institute
{mbiswas,fialm}@hbku.edu.qa, wajdi.zaghouani@northwestern.edu

Abstract

This paper presents the MAHED 2025 Shared Task on Multimodal Detection of Hope and Hate Emotions in Arabic Content, comprising three subtasks: (1) text-based classification of Arabic content into hate and hope,(2) multitask learning for joint prediction of emotions, offensive content, and hate speech and (3) multimodal detection of hateful content in Arabic memes. We provide three high-quality datasets totaling over 22,000 instances sourced from social media platforms, annotated by native Arabic speakers with Cohen's Kappa exceeding 0.85. Our evaluation attracted 46 leaderboard submissions from participants, with systems leveraging Arabic-specific pre-trained language models (AraBERT, MARBERT), large language models (GPT-4, Gemini), and multimodal fusion architectures combining CLIP vision encoders with Arabic text models. The best-performing systems achieved macro F1-scores of 0.723 (Task 1), 0.578 (Task 2), and 0.796 (Task 3), with top teams employing ensemble methods, class-weighted training, and OCR-aware multimodal fusion. Analysis reveals persistent challenges in dialectal robustness, minority class detection for hope speech, and highlights key directions for future Arabic content moderation research.

1 Introduction

Online platforms increasingly require robust systems to detect harmful and pro-social content. For Arabic, this need is compounded by dialectal diversity, code-switching, and multimodal formats (e.g., memes). Community evaluations have accelerated progress on Arabic toxicity: OSACT4 standardized offensive-language detection on Twitter, and OSACT5 extended to fine-grained hate speech, highlighting label imbalance and dialectal variation (Mubarak et al., 2020, 2022). New resources further enrich supervision, such as a multi-label Arabic corpus that jointly annotates offense, hate, emotion facets, sarcasm/humor, factuality, and perceived impact (Zaghouani et al., 2024) Surveys highlight key issues, such as implicit hate, target attribution and code-switching. They further emphasize the significance of Pretrained Language Models (PLMs), such as AraBERT and ARBERT/-MARBERT (Abdelsamie et al., 2024; Antoun et al., 2020; Abdul-Mageed et al., 2021). Beyond toxicity, detecting *hope speech* has emerged in LT-EDI shared tasks and offers complementary signals for safer moderation (Chakravarthi et al., 2022). Finally, research on multimodal harmful content shows that text-only or image-only models underperform on memes, motivating vision–language fusion; Arabic meme resources emphasize language-aware OCR and robust pipelines (Kiela et al., 2020; Alam et al., 2024b).

This paper presents the **MAHED 2025 Shared Task** on **M**ultitask **A**rabic **H**armful and **E**motional content **D**etection, comprising three subtasks: (i) **Text toxicity with hope**: classify text into *hate*, *hope*, or `not_applicable`; (ii) **Joint modeling**: simultaneously predict an emotion label with offensive and hate labels under an explicit hierarchy; and (iii) **Multimodal memes**: detect harmful content in image–text memes.[1]. The task is designed to investigate whether multitask and multimodal modeling improve robustness under dialectal variation, label skew, sarcasm, and noise from OCR text.

Contributions. We (1) define a three-part benchmark spanning text and memes; (2) detail datasets, label schemas, and evaluation protocols aligned with prior Arabic efforts and hope-speech literature; (3) release baseline training/evaluation code and configurations for Arabic PLMs and multimodal fusion; and (4) report results and error analyses across dialects and modalities.

[1] Exact data sources, splits, and scoring scripts are detailed in `https://github.com/marsadlab/MAHED2025Dataset.git`

Proceedings of The Third Arabic Natural Language Processing Conference, pages 560–574
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Related Work

Scope and definitions. We study two affective poles in Arabic: hate/offense (derogatory, dehumanizing, or abusive content) and hope (constructive, prosocial, future-oriented encouragement). We cover social media text and image memes, and acknowledge Arabic-specific challenges such as dialectal variability and code switching (Arabizi). This section positions MAHED with respect to Arabic hate/offense and hope in text, joint modeling with emotions, and multimodal detection in memes.

Arabic hate and offensive language in text. Community evaluations standardized tasks and metrics, accelerating progress. Mubarak et al. (2020) introduced Arabic offensive language detection on Twitter, and Mubarak et al. (2022) extended to finer-grained hate targets, highlighting dialectal variability and class imbalance. Beyond shared tasks, Zaghouani et al. (2024) released a 15,965 tweet multi label dataset (offense, hate, emotion facets, sarcasm/humor, factuality, perceived impact), where AraBERT style encoders outperform classical baselines; a recent survey synthesizes methods, datasets, and open challenges—including implicit hate, target attribution, and code switching—informing MAHED's taxonomy and evaluation (Abdelsamie et al., 2024). Strong Arabic PLMs such as AraBERT and ARBERT/MARBERT remain standard encoders for social media classification (Antoun et al., 2020; Abdul-Mageed et al., 2021). Overall, text-only Arabic toxicity is relatively mature, while gaps persist in dialectal robustness, implicit hate, and correlated labels under class imbalance, which MAHED targets explicitly.

Hope speech and prosocial content. Hope speech is increasingly treated as a distinct class of constructive and supportive online content in the LT and EDI communities. Shared tasks report that transformer-based models consistently outperform classical approaches for hope speech classification (Chakravarthi et al., 2022). Beyond shared tasks, work on Urdu social media shows that transformer models obtain the top macro F1 for multi-class hope and hopelessness, and that careful annotation guidelines help capture nuanced expressions of hope (Balouchzahi et al., 2025). Complementary psycholinguistic analyses indicate that hope speech displays distinctive cognitive, emotional, and communicative profiles, and that tree boosting methods such as LightGBM and CatBoost can be competitive for type-level hope classification when tuning is performed (Arif et al., 2024). Theory and experiments in social psychology connect specific emotions to prosocial behavior: emotions such as hope and gratitude can motivate helping through both intrapersonal and interpersonal pathways (van Kleef and Lelieveld, 2022), and hopeful reappraisal in distressing contexts has been shown to increase charitable giving (Brethel-Haurwitz et al., 2020). Together, these results support modeling hope as a separate target alongside hate or offense in Arabic, to avoid conflation with generic positivity and to enable evaluation of prosocial language in culturally specific settings.

Emotion analysis in Arabic. Arabic emotion analysis has progressed in both text and speech, enabling fine-grained affect modeling. For social content, resources such as ArPanEmo support recognition of multiple emotions, plus neutral, and allow multi-class setups (Althobaiti, 2023). In speech, the King Saud University Emotions corpus and related datasets demonstrate that speaker gender, emotion type, and their interaction affect perception and recognition, and they provide a basis for statistical and perceptual analyses (Meftah et al., 2018, 2021). Studies on Arabic dialects report strong performance with standard classifiers, as well as with prosodic and spectral features. For example, support vector machines provide about 77 percent accuracy on Saudi dialect data (Aljuhani et al., 2021), long-term average spectrum and wavelet features yield improvements for Egyptian Arabic (Abdel-Hamid, 2020), and multistage classification schemes offer reasonable gains (Poorna and Nair, 2019). Earlier studies based on TV show speech, along with subsequent surveys, highlight the consistent roles of pitch, intensity, speaking rate, and mel-frequency cepstral coefficients (MFCCs), while also underscoring the open challenges of achieving cross-speaker and cross-dialect generalization (Klaylat et al., 2018; Meddeb et al., 2017; Nasr et al., 2024). Evidence from perceptual research indicates that prosody and lexical semantics contribute through separate yet intertwined channels, with prosodic dominance often observed (Ben-David et al., 2016). In parallel, corpus-based studies of Arabic vocabulary in religious texts highlight a wide lexical space for emotional expression, underscoring the need for culturally informed annotation and modeling choices (Salsabila et al., 2024). These findings motivate the integration of emotion signals with toxicity and prosociality labels. Additionally, in order to address label imbalance and better capture minority classes such as hope, multi-label or ordinal objectives can be adopted.

Multitask and multi-label modeling. Given correlated labels (for example, hate ⇒ offense; emotion ↔ toxicity), joint learning can improve minority classes via shared representations. In Arabic, multitask architectures that combine offense/hate with sentiment or related signals improved robustness on OSACT style data (Abu Farha and Magdy, 2020; Djandji et al., 2020). MAHED follows this paradigm in Task 2 by jointly predicting emotions, offensive content, and hate under an explicit label hierarchy.

Multimodal harmful content and Arabic memes. The Hateful Memes benchmark demonstrated the insufficiency of unimodal baselines and popularized vision–language fusion (Kiela et al., 2020). Subsequent efforts, such as MultiOFF and the SemEval 2022 MAMI task, further highlighted the benefits of fusing text and image and incorporating subtype labels (Suryawanshi et al., 2020; Fersini et al., 2022). For Arabic, Alam et al. (2024b) introduced ArMeme, a manually annotated meme dataset targeting propagandistic techniques, and established text-image fusion as essential baselines for Arabic script and domains. Building on this trend, MAHED extends the scope to Arabic memes by evaluating OCR-aware text–image fusion for both hate and hope, while leaving speech and video analysis out of scope for this edition.

Summary and link to design. From 2020 to 2025, Arabic hate/offense matured via shared tasks and PLMs, affect resources expanded, and hope remained comparatively under-resourced in Arabic. Multitask and multimodal fusion approaches have been consistently beneficial. In response, MAHED unifies hate, offense, and hope annotations for Arabic text, investigates joint learning with emotions to improve the representation of minority classes, and extends its scope to OCR-aware text–image fusion, with particular attention to dialect variation and code-switching.

3 Tasks and Datasets

The MAHED shared task consists of three subtasks: **(1)** Text-based Hope and Hate Speech Classification, **(2)** Multitask Learning for Emotion, Offensive Content, and Hate Detection, and **(3)** Multimodal Hateful Meme Detection. All content in the related datasets was sourced from public social media platforms, anonymized to protect user privacy, and annotated by native Arabic speakers. The annotation process achieved a high inter-annotator agreement, with a Cohen's Kappa score exceeding 0.85, indicating strong consistency among annotators.

Data Partition	Label	Count	Dist.
Train (6,890)	Hate	1,301	18.9%
	Hope	1,892	27.5%
	NA	3,697	53.7%
Dev (1,476)	Hate	261	17.7%
	Hope	409	27.7%
	NA	806	54.6%
Test (1,477)	Hate	287	19.4%
	Hope	422	28.6%
	NA	768	52%

Table 1: Distribution of class labels in the Task 1 dataset. NA: not_applicable

3.1 Task 1 : Text-based Hope and Hate Speech Classification

Task: The objective of the first task is to develop a model that classifies Arabic text into one of three categories: *"hate"*, *"hope"*, and *"not_applicable"*. In this context, *hate* refers to expressions that contain offensive, discriminatory, or harmful language directed toward individuals or groups based on features such as religion, nationality, ethnicity, or other protected characteristics. *Hope* refers to expressions of positive emotional content, including aspirational, motivational, or future-oriented messages, as well as statements that convey optimism, gratitude, or encouragement. The *not_applicable* category includes all remaining cases that do not contain explicit hate or hope content.

Dataset: The dataset used for this task consists of 9,843 high-quality Arabic text instances that have been carefully prepared for classification into the *"hate"*, *"hope"*, and *"not_applicable"* categories. The data is divided into three subsets: 6,890 samples for training, 1,476 for validation, and 1,477 for testing. The dataset have been obtained from the combination of three high quality datasets (Zaghouani et al., 2024; Zaghouani and Biswas, 2025b,a). Table 1 presents the label distribution across the training, validation, and test sets, reporting both the number of instances in each category and their relative proportions.

3.2 Task 2: Multitask Learning for Emotion, Offensive Content, and Hate Detection

Task. The second task addresses multitask learning for joint emotion, offensive language, and hate speech detection in Arabic text. The objectives of this task are (i) predicting a single emotion label from a predefined list of 12 emotions (*neutral, anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust*), (ii) determining whether the text is offensive (*yes/no*), and (iii) if offensive, deciding if the text is hate speech (*hate* vs. *not_hate*). This order reflects the hierarchical relationship between offensiveness and hate since all hate speech is offensive, but not all offensive content is hate speech. Specifically, texts labeled as *hate* contain offensive content directed at an identity group (e.g., religion, nationality, ethnicity, or gender). In contrast, texts labeled as *not_hate* may also be offensive but do not target specific identities, such as instances of casual or profane language without identity-based targeting.

Dataset. The dataset for this task comprises 8,515 high-quality annotated Arabic text instances, prepared for joint classification of emotions, offensive language, and hate speech. Three high quality data sources were used for curation of this shared task datasets (Zaghouani et al., 2024; Zaghouani and Biswas, 2025b,a). It is divided into three subsets: 5,960 samples for training, 1,277 for validation, and 1,278 for testing. Each instance is labeled with three layers of information aligned with the task objectives: (i) one emotion from the 12 categories, (ii) an offensiveness label (*yes/no*), and (iii) for offensive texts, a hate label distinguishing between *hate* and *not_hate*. Table 2 summarizes the distribution of these label categories across the training, validation, and test sets.

Label	Train	Val	Test
Emotion			
Neutral	661	137	128
Anger	1,551	331	327
Anticipation	491	121	120
Disgust	777	153	167
Fear	53	9	13
Joy	533	120	135
Love	593	135	117
Optimism	419	88	79
Pessimism	194	54	39
Sadness	335	54	68
Surprise	143	28	33
Confidence (Trust)	210	47	52
Offensive			
Yes	1,744	363	370
No	4,216	914	908
Hate (if offensive)			
Hate	303	68	69
Not hate	1,441	294	301
Total	**5,960**	**1,277**	**1,278**

Table 2: Label distribution in the Task 2 dataset across training, validation, and test splits.

3.3 Task 3 : Multimodal Hateful Meme Detection

Task. The objective of this subtask is to *determine whether a meme—comprising both textual and visual content—is hateful or not*, formulated as a binary classification problem. Participants were allowed to adopt any experimental setup, leveraging text-only, image-only, or multimodal approaches.

Dataset. For this subtask, the dataset is derived from prior work (Hasanain et al., 2024; Alam et al., 2024c,a) and comprises 3,562 memes, including the final evaluation test set. These memes were collected from diverse social media platforms such as Facebook, Twitter, Instagram, and Pinterest. The textual content within the memes was extracted using an off-the-shelf OCR tool[2], followed by manual post-editing to ensure accuracy.

Hateful meme annotations for the training and development sets were obtained through a hybrid approach, combining multiple large language models (LLMs) replicating human annotation approaches. The test set (referred to as dev-test) was fully human-annotated. For the shared task, we additionally constructed a new test split, adhering to the data collection methodology and annotation guidelines described in (Alam et al., 2024c).

4 Results

This section reports the leaderboard results for each of the three subtasks, including the team rankings and their corresponding Macro F1-scores.

4.1 Task 1

Task 1 received a total of 28 submissions. The baseline system, a BERT-based model, achieved a Macro F1-score of 0.53, providing a reference point for evaluating participant systems. As shown in Table 3, *HTU* (Saleh and Biltawi, 2025) achieved

[2] https://github.com/JaidedAI/EasyOCR

the highest performance with a Macro F1-score of 0.723. Their system combined multiple Arabic models (ArabicDeBERTa-DA, BERT-MSA, MARBERTv2) in an ensemble, which allowed them to capture variation across dialects and improved robustness. *NYUAD* (AlDahoul and Zaki, 2025), the second-ranked team with 0.721 F1-score, leveraged large language models by fine-tuning GPT-4o-mini and Gemini Flash 2.5 alongside Google text embeddings with an SVM classifier, and fused predictions through majority voting, which helped them handle subjective and dialectal confusions. *AAA* (Elzainy et al., 2025) and *NguyenTriet* (Nguyen and Dang, 2025a) shared third place with an F1-score of 0.707. AAA systematically evaluated multiple transformer encoders and found that MARBERT was the most effective. NguyenTriet, by contrast, used a carefully preprocessed dataset and built an ensemble of Arabic-specific BERT encoders with soft-voting fusion.

LoveHeaven (Nguyen and Dang, 2025b) achieved strong results (0.703) by ensembling AraBERT-Twitter variants and incorporating attention-based features. *IRIT_HOPE* (Moudjari et al., 2025) also ranked among the top systems (with 0.701), combining token-level augmentation with pragmatic features derived from multiple sources (MAHED, MLMA, and synthetic data). *phucclone** likewise delivered a competitive performance, securing a place within the top seven.

Beyond the top-performing group, several other teams achieved competitive results. For instance, *novatriee**, *CUET_Zahra_Duo* (Alam et al., 2025) (which fine-tuned AraBERTv2-large with optimized early stopping), *ahmedabdou** and *TranTranUIT* (Tran and Dang, 2025), all scored near 0.69. *TranTranUIT* focused on dialect sensitivity and cross-lingual generalization, applying extensive data augmentation strategies including backtranslation , EDA-based transformations, and noise reduction. They fine-tuned AraBERTv2, AraBERT-Twitter, and XLM-RoBERTa, combining them in a soft-voting ensemble.

Teams clustered in the 0.64–0.69 range included *SmolLab_SEU* (Rahman et al., 2025), which experimented with several Arabic-native and multilingual transformers, and *Quasar* (Chowdhury and Chowdhury, 2025), which combined text normalization with data augmentation and large models. Other teams in this group were *CIC-NLP* (Obiadoh et al., 2025), *ANLPers* (Yasser et al., 2025), *sudo_apt**, *Muhammad Annas Shaikh**, *michaelibrahim**, *minhtriet**, *nguyenminhtriet**, *Baoflowin502* (Bao and Thin, 2025), *KALAM* (Hameed and Al-Fuqaha, 2025), *AraNLP* (Khalil and El-Kassas, 2025), and *turabusmani**. The lowest-ranked group — including *ANLP-UniSo* (El Abed et al., 2025), *REGLAT* (Ashraf et al., 2025), *shadmansaleh**, and *AyahVerse* (Rashid and Khalil, 2025) — scored below 0.60.

Rank	Team	F1-score
1	**HTU**	**0.723**
2	**NYUAD**	**0.721**
3	**AAA**	**0.707**
3	**NguyenTriet**	**0.707**
4	LoveHeaven	0.703
5	IRIT_HOPE	0.701
6	phucclone*	0.700
7	novatriee*	0.698
8	CUET_Zahra_Duo	0.695
9	ahmedabdou*	0.695
10	trantranuit	0.694
11	SmolLab_SEU	0.682
12	Quasar	0.674
13	CIC-NLP	0.673
14	ANLPers	0.672
15	sudo_apt*	0.671
16	Muhammad Annas Shaikh*	0.669
17	michaelibrahim*	0.665
18	minhtriet*	0.659
18	nguyenminhtriet*	0.659
19	Baoflowin502	0.651
20	KALAM	0.650
20	AraNLP	0.650
21	turabusmani*	0.647
22	ANLP-UniSo	0.595
23	REGLAT	0.579
baseline	Baseline model	0.53
25	shadmansaleh*	0.483
25	AyahVerse	0.481

*The corresponding papers were not submitted.

Table 3: Task 1 results with team rankings

4.2 Task 2

Task 2 received a total of 11 submissions. The baseline system, built with an AraBERT model, achieved a Macro F1-score of 0.50. As shown in Table 4, *NYUAD* ranked first with a Macro F1-score of 0.578. Their system trained three fine-tuned GPT-4o-mini models, each specialized for emotion, offensive, and hate detection sub-tasks. They further addressed class imbalance by oversampling the "hate" class fivefold. *NguyenTriet*, in second place with 0.553, developed a hierarchical cascade architecture where predictions from emotion classification were fed into offensiveness detection, which in turn informed hate detection. They relied on ensembling MARBERTv2 and AraBERT-Twitter with soft voting at each stage. Rigorous text normalization (emoji demojization, diacritic removal, URL/stopword filtering) and class-weighted training with cosine learning-rate scheduling improved their ability to handle imbalance and dialectal variation. *HTU* placed third with 0.535, proposing a Retrospective Reader with an ALBERT approach. Their system first produced an initial prediction and then used retrospective verification to refine the classification, which helped reduce false positives. *CUET_823* (Dhar and Mallik, 2025), ranking fourth with 0.518, applied Meta-Llama-3.1-8B with instruction tuning and quantization (LoRA + 4-bit) for efficiency. They used a two-stage prompt-based approach that enabled zero- and few-shot adaptability. Finally, *SmolLab_SEU* finished in the top five with 0.514, building three separate classifiers for emotion, offensive, and hate detection using a wide range of pretrained models (MARBERTv2, ARBERTv2, AraBERTv2-large, QARiB, XLM-RoBERTa-large, mDeBERTaV3-base, DistilBERT-base). The remaining teams, including Quasar, deleted_user_25186*, KALAM, turabusmani*, MultiMinds (Debnath et al., 2025), and ashfaq*, scored between 0.33 and 0.48. These systems struggled with borderline distinctions between offensive and hate, as well as imbalanced data, highlighting the difficulty of this subtask compared to Task 1.

Rank	Team	F1-score
1	**NYUAD**	**0.578**
2	**NguyenTriet**	**0.553**
3	**HTU**	**0.535**
4	**CUET_823**	**0.518**
5	**SmolLab_SEU**	**0.514**
baseline	Baseline model	0.50
6	Quasar	0.480
7	deleted_user_25186*	0.459
8	Kalam	0.434
9	turabusmani*	0.398
10	MultiMinds	0.349
11	ashfaq*	0.336

*The corresponding papers were not submitted.

Table 4: Task 2 results with team rankings and Macro F1-scores

4.3 Task 3

Task 3 received a total of 7 submissions. The baseline multimodal hateful-meme detection system obtained a Macro F1-score of 0.70. As shown in Table 5, *NYUAD* achieved the best performance with a Macro F1-score of 0.796, the highest across all subtasks. The next two teams, *yassirEA* (0.750) (El Attar, 2025) and *Araminds* (0.744) (Zaytoon et al., 2025), also performed strongly, both surpassing 0.74. *thinkingNodes* (Safwan, 2025) followed in fourth place with 0.718, while *Muhammad Anas Shaikh** and *joy_2004114* (Das et al., 2025) obtained mid-range scores of 0.684 and 0.629, respectively. *MultiMinds* ranked last with 0.497.

Rank	Team	F1-score
1	**NYUAD**	**0.796**
2	**yassirEA**	**0.750**
3	**Araminds**	**0.744**
4	**thinkingNodes**	**0.718**
baseline	Baseline Model	0.70
5	Muhammad-Annas Shaikh*	0.684
6	joy_2004114	0.629
7	MultiMinds	0.497

*The corresponding papers were not submitted.

Table 5: Task 3 results with team rankings and Macro F1-scores

5 System Description

5.1 Data Preprocessing Techniques

The most common preprocessing steps applied by teams are summarized below:

- **Tokenization** (8 teams: SmolLab_SEU, AAA, KALAM, AraNLP, HTU, REGLAT, MultiMinds, NYUAD): Segmenting text into tokens for compatibility with deep learning models.
- **Remove URLs** (6 teams: NguyenTriet, SmolLab_SEU, KALAM, AraNLP, REGLAT, MultiMinds): Eliminating hyperlinks to reduce noise in social media text.
- **Remove Mentions/Hashtags** (5 teams: NguyenTriet, SmolLab_SEU, REGLAT, LoveHeaven, Araminds): Stripping social media markers that encode metadata rather than content.
- **Lowercasing/Normalization** (4 teams: NguyenTriet, SmolLab_SEU, KALAM, MultiMinds): Standardizing case and script

forms to reduce vocabulary redundancy.

- **AraBERT Preprocessing** (4 teams: AraNLP, CIC-NLP, LoveHeaven, AyahVerse): Using an Arabic-specific pipeline for diacritic removal, normalization, and script unification.

5.2 Feature Engineering

Text-based Tasks (Task 1 and Task 2) For the text-only tasks, teams employed both traditional vectorization methods and deep contextual embeddings:

- Google text embeddings with SVM (NYUAD): Used pretrained Google text embeddings as input to an SVM classifier, providing a strong baseline with fixed semantic representations.
- Ensemble of Arabic-specific BERT encoders (NguyenTriet): Combined outputs from MARBERTv2 and related encoders to improve robustness across dialectal variation.
- TF–IDF and Embedding-based Features (KALAM, REGLAT): Leveraged classical TF–IDF along with embeddings from AraBERT, CAMeL-BERT, and MARBERT; in some cases, attention-based features were added to capture contextual cues.
- Bag-of-Words and Morphological Features (trantranuit, CIC-NLP): Applied n-gram BoW features, enriched with morphological features such as POS tags, verb patterns, and affixes.
- Attention-based Features (KALAM, Muhammad Annas Shaikh, LoveHeaven): Extracted attention weights from transformer models as features, highlighting salient contextual dependencies.
- Augmentation and Scaling (IRIT_HOPE): Introduced token-level augmentation and normalized log feature scaling to improve robustness and feature balance.
- Linguistic Features and Normalization (CIC-NLP, Quasar): Integrated handcrafted linguistic signals and normalization of diacritics to reduce noise in Arabic text.

Multimodal Task (Task 3) For the multimodal setting (image + text), teams explored fusion strategies combining visual and textual embeddings:

- Google Multimodal Embeddings: Used 512-dimensional embeddings for both image and text, fused via element-wise averaging or concatenation.
- Pretrained Encoders with Fusion (CLIP, MARBERT): Extracted features from CLIP-ViT (vision) and MARBERT (text), projecting them into a shared space and applying cross-attention or gated fusion strategies.
- Dual-encoder Architectures: Combined text and image encoders with late fusion, optimizing with binary cross-entropy and contrastive losses to align modalities.
- Hybrid Fusion Models: Used CLIP ViT-B/32 features with text embeddings (e.g., DistilBERT) and fused them using cross-attention modules.
- Advanced Fusion (MARBERTv2 + CLIP ViT-L/14): Explored multiple fusion mechanisms, including transformers, early concatenation, bilinear pooling, and cross-attention for joint representation learning.

5.3 Data Augmentation

Text-based Tasks (Task 1 and Task 2) For the text-only tasks, teams experimented with different augmentation strategies, although many reported limited or no improvement.

- Synonym replacement and back-translation were applied to increase lexical diversity, though in some cases they did not yield performance gains.
- Synthetic Minority Over-sampling Technique (SMOTE) and oversampling were used to generate synthetic minority samples, balancing class distributions and reducing bias in training data.
- Easy Data Augmentation (EDA) techniques such as random insertion, swapping, deletion, and synonym replacement were employed to expand the dataset with simple transformations.
- Bigram augmentation and contextual embeddings were explored to introduce variation at both the lexical and semantic levels.
- Some teams leveraged external synthetic and multilingual datasets (e.g., MAHED, MLMA) to supplement training and cover dialectal variation.

Multimodal Task (Task 3) In the multimodal meme classification task, augmentation targeted both text and image modalities.

- Oversampling of hate memes was performed up to nine times to alleviate class imbalance and strengthen minority-class learning.
- Image-based augmentation included rotation, scaling, perspective shifts, color jitter, gamma

Type	Model	T1	T2	T3	Key advantage
Transformer	AraBERT/v2	✓	✓		Arabic morphology
	MARBERT/v2	✓	✓	✓	Noisy social text
	CAMeL-BERT	✓			Robust baseline
	QARiB	✓			News/social adapted
	XLM-RoBERTa	✓	✓		Multilingual
	DistilBERT	✓		✓	Lightweight
	DeBERTa variants	✓			Better attention
Vision	CLIP (ViT)			✓	Vision-text align
	ResNet/ResNeXt			✓	Visual backbone
LLM/VLM	GPT-4	✓	✓	✓	Few-shot learning
	Gemini	✓	✓	✓	Multimodal reason
	LLaMA			✓	Finetuned branch
	Gemma			✓	Compact VLM
	Qwen			✓	Multilingual VLM

Table 6: Model families used across tasks

correction, noise, blurring, distortions, shadows, fog effects, and crop–resize operations.

- Text within memes was augmented using OCR-based extraction followed by synonym replacement, character-level dropout, and back-translation between Arabic and English.
- Some teams focused augmentation specifically on hate-class examples, ensuring that rare cases were better represented in multimodal training.

5.4 Model Usages Across Tasks

Task 1: Text-based Hope and Hate Speech Classification. Teams primarily used Arabic-centric transformers (AraBERT, MARBERT, CAMeL-BERT, QARiB, XLM-R) to obtain context-aware sentence embeddings robust to morphology, code-mixing, and informal orthography. These encoders work well for short, noisy social posts where pragmatic cues and dialectal markers are crucial. LLMs (e.g., GPT-4, Gemini) appeared as auxiliary backbones or zero/few-shot components, valued for broad world knowledge and flexible prompting when labeled data are limited.

Task 2: Multitask Emotion/Offense/Hate. A shared transformer encoder with lightweight task heads provides a compact way to model related label spaces, enabling representation sharing across emotion, offensive content, and hate signals. This setup simplifies training pipelines and reduces overfitting via shared inductive biases; LLMs help unify task instructions and can serve as promptable controllers for multi-objective finetuning.

Task 3: Multimodal Hateful Meme Detection. Vision–language stacks (CLIP/ViT + Arabic text encoders) align image and text into a shared semantic space so that cross-modal cues—caption sarcasm, visual symbols, and text overlays—can be interpreted jointly. LLM/VLM components (Gemini, LLaMA, Gemma, Qwen) are useful where reasoning over both modalities or following structured prompts improves recognition of subtle or template-driven hateful content.

5.5 Training Configurations and Rationale

Drop-in Recipes (Space-Efficient, Reproducible)

Recipe: Text Hope/Hate

Encoder: AraBERTv2 or MARBERTv2; max length 256; batch 16; LR $2{\times}10^{-5}$ (AdamW, WD 0.01), 10% warmup, cosine decay, FP16, grad clip 1.0. Class-weighted CE; early stopping on macro-F1 (patience 3); 5-fold stratified CV; select best checkpoint by macro-F1.

Recipe: Multitask (Emotion/Offense/Hate)

Shared encoder (AraBERT/MARBERT) with multi-head classifiers; batch 16 (grad accum 2); LR $1{\times}10^{-5}$; warmup 10%, cosine schedule; FP16. Class-weighted CE; early stopping on macro-F1. Tune per-head dropout/epochs via Optuna; optional LR multiplier ($\approx$1.8) for heads.

Recipe: Multimodal Memes

Text: MARBERTv2 [CLS] or DistilBERT tokens; Image: CLIP ViT-B/32 (or ViT-L/14). Project to 512-d; fuse by concatenation or cross-attention. Batch 16–32 (per-device 2–4 for large VLMs); LR text/vision $2{\times}10^{-5}$, fusion head $1{\times}10^{-3}$; AdamW (WD 10^{-4}), linear or cosine schedule; FP16, grad clip 1.0. Loss: weighted BCE/CE, focal-loss trial. Early stopping with patience 5–15; oversample minority class.

Use Cases

- *Macro-F1 selection, class-weighted losses, and oversampling* address severe label imbalance (hate/hope and multimodal memes), prioritizing minority-class recall without inflating accuracy.

Task	Typical Backbones	Epochs	Batch Size	Seq. Length	Learning Rate	Optimizer & Strategy
Hope/Hate (Text)	AraBERT, MARBERT, CAMeLBERT, XLM-RoBERTa, QARiB, ArabicDeBERTa	2–10 (ES:3–5)	16–32	128–256	310^{-6}–110^{-5}	AdamW, cosine scheduler, FP16
Multitask (Text)	AraBERT, MARBERT variants	3–10	8–16	128	110^{-5}–210^{-5}	AdamW, warmup/cosine, FP16
Multimodal Memes	CLIP ViT + MARBERT; VLMs (Gemma, Qwen, Paligemma)	5–40 (ES) VLM:10	16–32 2–4 (VLM)	Variable	Text 110^{-5} Vision 210^{-5} VLM 510^{-6}	AdamW, gradient clip 1.0, cross-attention fusion

Table 7: Typical training settings distilled from submitted systems across tasks. ES = early stopping, VLM = vision-language model.

- *Warmup + cosine/linear schedules with AdamW* stabilize finetuning of large encoders and prevent early-step divergence; *weight decay and dropout* regularize under limited data.
- *FP16 and gradient clipping* improve memory efficiency and prevent exploding gradients, which is critical in multimodal or multitask finetuning.
- *Shared encoders with task heads (multitask)* reuse domain signals (emotion, offense, hate) and conserve parameters; LR multipliers let heads adapt faster without overfitting the encoder.
- *CLIP+Arabic encoders with projection/fusion* capture cross-modal interactions in memes; aligning to a 512-d shared space simplifies fusion while retaining modality-specific strengths.
- *CV and Optuna* provide robust, reproducible hyperparameters without exhaustive grids; reporting the validation macro-F1 criterion ensures consistent model selection.

6 Conclusions and Future Work

The MAHED 2025 shared task establishes comprehensive benchmarks for Arabic content moderation across textual and multimodal formats. With 46 participating teams, the evaluation demonstrates consistent improvements over baselines, achieving macro F1-scores of 0.723 (Task 1), 0.578 (Task 2), and 0.796 (Task 3). Top systems leveraged Arabic-specific PLMs (AraBERT, MARBERT), ensemble methods, and OCR-aware multimodal fusion.

Key Challenges: Our analysis reveals persistent limitations: (i) dialectal robustness gaps of up to 34% in error cases, with Gulf and Levantine expressions frequently misclassified; (ii) minority class detection difficulties, particularly for hope speech (average recall: 0.52); (iii) OCR noise contributing to 28% of multimodal errors; and (iv) Task 2's hierarchical multitask complexity, where conflicting optimization pressures across emotion, offense, and hate detection yielded the lowest performance (0.578 F1).

Future Directions: Critical research priorities include: dialect-invariant representations through cross-dialectal augmentation and adversarial training; culturally-grounded hope speech annotation with contrastive learning objectives; Arabic-specific scene text recognition for stylized fonts; and uncertainty-aware multitask architectures. Evaluation methodology should incorporate dialectal breakdowns, calibration analysis, and fairness auditing.

Impact: The released datasets (22,000+ instances, Cohen's Kappa >0.85), baseline implementations, and comprehensive analysis provide a reproducible foundation for Arabic content safety research. While significant progress was demonstrated, the identified challenges underscore the need for culturally-informed approaches that address Arabic's unique linguistic and cultural characteristics.

7 Limitations

The MAHED shared task has several inherent constraints: (i) focus on social media data excludes formal Arabic domains; (ii) binary hope/hate categories oversimplify the prosocial-harmful spectrum; (iii) hierarchical multitask design in Task 2 introduces conflicting optimization pressures; (iv) OCR-dependent multimodal processing creates systematic extraction errors; and (v) annotation guidelines may not fully capture dialectal and cultural diversity across Arabic-speaking regions.

Acknowledgments

We thank all participating teams for their contributions, as well as the organizers of the Arabic NLP 2025 Conference, and the broader community for their support. The authors thank our annotation team for ensuring the authenticity and appropriateness of our resources. The contribution of this work is supported by the NPRP grant 14C-0916-210015 from the Qatar National Research Fund, part of the Qatar Research Development and Innovation Council (QRDI). The findings achieved herein are solely the responsibility of the authors.

References

Lamiaa Abdel-Hamid. 2020. Egyptian arabic speech emotion recognition using prosodic, spectral and wavelet features. *Speech Communication*, 122:19–30.

Mahmoud Mohamed Abdelsamie, Shahira Shaaban Azab, and Hesham A. Hefny. 2024. A comprehensive review on arabic offensive language and hate speech detection on social media: methods, challenges and solutions. *Social Network Analysis and Mining*, 14(1):111.

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.

Ibrahim Abu Farha and Walid Magdy. 2020. Multitask learning for Arabic offensive language and hate-speech detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 86–90, Marseille, France. European Language Resource Association.

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmad, Md. Arid Hasan, and Maram Hasanain. 2024b. ArMeme: Propagandistic content in Arabic memes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21071–21090, Miami, Florida, USA. Association for Computational Linguistics.

Firoj Alam, Abul Hasnat, Fatema Ahmed, Md Arid Hasan, and Maram Hasanain. 2024c. ArMeme: Propagandistic content in arabic memes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Walisa Alam, Mehreen Rahman, Shawly Ahsan, and Mohammed Moshiul Hoque. 2025. Cuet_zahra_duo@mahed 2025: Hate and hope speech detection in arabic social media content using transformer. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Nouar AlDahoul and Yasir Zaki. 2025. Nyuad at mahed shared task: Detecting hope, hate, and emotion in arabic textual speech and multi-modal memes using large language models. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

R. H. Aljuhani, A. Alshutayri, and H. Alahdal. 2021. Arabic speech emotion recognition from saudi dialect corpus. *IEEE Access*, 9:127081–127085.

Maha Jarallah Althobaiti. 2023. An open-source dataset for arabic fine-grained emotion recognition of online content amid covid-19 pandemic. *Data in Brief*, 51:109745.

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.

Muhammad Arif, Moein Shahiki Tash, Ainaz Jamshidi, Fida Ullah, Iqra Ameer, Jugal Kalita, Alexander Gelbukh, and Fazlourrahman Balouchzahi. 2024. Analyzing hope speech from psycholinguistic and emotional perspectives. *Scientific Reports*, 14(1):23548.

Nsrin Ashraf, Mariam Labib, Tarek Elshishtawy, and Hamada Nayel. 2025. Reglat at mahed shared task: A hybrid ensemble-based system for arabic hate speech detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

F. Balouchzahi and 1 others. 2025. Urduhope: Analysis of hope and hopelessness in urdu texts. *Knowledge-Based Systems*, 308:112746.

Nguyen Minh Bao and Dang Van Thin. 2025. Baoflowin502 at mahed shared task: Text-based hate and hope speech classification. In *Proceedings of the*

Third Arabic Natural Language Processing Conference (ArabicNLP 2025), Suzhou, China. Association for Computational Linguistics.

Boaz M. Ben-David, Nandini Multani, Vered Shakuf, Frank Rudzicz, and Pascal H. H. van Lieshout. 2016. Prosody and semantics are separate but not separable channels in the perception of emotional speech: Test for rating of emotions in speech. *Journal of Speech, Language, and Hearing Research*, 59(1):72–89.

Kristin M. Brethel-Haurwitz, Maria Stoianova, and Abigail A. Marsh. 2020. Empathic emotion regulation in prosocial behaviour and altruism. *Cognition and Emotion*, 34(8):1532–1548.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, Subalalitha Chinnaudayar Navaneethakrishnan, John P. McCrae, Miguel Ángel García, Salud María Jiménez-Zafra, Rafael Valencia-García, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Daniel García-Baena, and José Antonio García-Díaz. 2022. Overview of the shared task on hope speech detection for equality, diversity and inclusion. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 378–388, Dublin, Ireland. Association for Computational Linguistics.

Md Sagor Chowdhury and Adiba Fairooz Chowdhury. 2025. Quasar at mahed shared task : Decoding emotions and offense in arabic text using llm and transformer-based approaches. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Joy Das, Alamgir Hossain, and Mohammed Moshiul Hoque. 2025. joy_2004114 at mahed shared task : Filtering hate speech from memes using a multimodal fusion-based approach. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Riddhiman Swanan Debnath, Abdul Wadud Shakib, and Md Saiful Islam. 2025. Multiminds at mahed 2025: Multimodal and multitask approaches for detecting emotional, hate, and offensive speech in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Ratnajit Dhar and Arpita Mallik. 2025. Cuet-823 at mahed 2025 shared task: Large language model-based framework for emotion, offensive, and hate detection in arabic. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Marc Djandji, Fady Baly, Wissam Antoun, and Hazem Hajj. 2020. Multi-task learning using AraBert for offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 97–101, Marseille, France. European Language Resource Association.

Yasmine El Abed, Mariem Ben Arbia, Saoussen Ben Chaabene, and Omar Trigui. 2025. Anlp-uniso at mahed shared task: Detection of hate and hope speech in arabic social media based on xlm-roberta and logistic regre. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Yassir El Attar. 2025. Yassirea at mahed 2025: Fusion-based multimodal models for arabic hate meme detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Ahmed Elzainy, Hazem Abdelsalam, Ahmed Samir, and Mohamed Amin. 2025. Aaa at mahed shared task: A systematic encoder evaluation for arabic hope and hate speech classification. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. 2022. SemEval-2022 task 5: Multimedia automatic misogyny identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 533–549, Seattle, United States. Association for Computational Linguistics.

Saad Hameed and Ala Al-Fuqaha. 2025. Kalam at mahed shared task 2025: Transformer-based approaches for arabic sentiment classification and stance detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Maram Hasanain, Md. Arid Hasan, Fatema Ahmed, Reem Suwaileh, Md. Rafiul Biswas, Wajdi Zaghouani, and Firoj Alam. 2024. ArAIEval Shared Task: Propagandistic techniques detection in unimodal and multimodal arabic content. In *Proceedings of the Second Arabic Natural Language Processing Conference*, Bangkok. ACL.

Enas A. Hakim Khalil and Wafaa S. El-Kassas. 2025. Aranlp at mahed 2025 shared task: Using arabert for text-based hate and hope speech classification. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: detecting hate speech in multimodal memes.

In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Samira Klaylat, Zeina Osman, and Rached Zantout. 2018. Emotion recognition in arabic speech. *Analog Integrated Circuits and Signal Processing*, 96(2):185–198.

Mohamed Meddeb, Rima Malka, and Mohamed Ali Hammami. 2017. Building and analysing emotion corpus of the arabic speech. In *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 18–22.

Ali Hamid Meftah, Yousef Ajami Alotaibi, and Sid-Ahmed Selouani. 2018. Evaluation of an arabic speech corpus of emotions: A perceptual and statistical analysis. *IEEE Access*, 6:72845–72861.

Ali Hamid Meftah, Mohammad A. Qamhan, Yasser M. Seddiq, Yousef Ajami Alotaibi, and Sid-Ahmed Selouani. 2021. King saud university emotions corpus: Construction, analysis, evaluation, and comparison. *IEEE Access*, 9:54201–54219.

Leila Moudjari, Mélissa Hacene Cherkaski, and Farah Benamara. 2025. Descartes_hope at mahed shared task 2025: Integrating pragmatic features for arabic hope and hate speech detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Hamdy Mubarak, Hend Al-Khalifa, and Abdulmohsen Al-Thubaity. 2022. Overview of OSACT5 shared task on Arabic offensive language and hate speech detection. In *Proceedinsg 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 162–166, Marseille, France. European Language Resources Association.

Hamdy Mubarak, Kareem Darwish, Walid Magdy, Tamer Elsayed, and Hend Al-Khalifa. 2020. Overview of OSACT4 Arabic offensive language detection shared task. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 48–52, Marseille, France. European Language Resource Association.

L. Nasr, M. Hani, A. Harkous, Y. Al Khalil, A. Abou Daya, H. Hajj, and R. El-Khoury. 2024. Survey on arabic speech emotion recognition. *International Journal of Speech Technology*. Online first.

Minh Triet Nguyen and Van Thin Dang. 2025a. Nguyentriet at mahed shared task: Ensemble of arabic bert models with hierarchical prediction and soft voting for text-based hope and hate detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Thien Bao Nguyen and Van Thin Dang. 2025b. Loveheaven at mahed 2025: Text-based hate and hope speech classification using arabert-twitter ensemble. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

A. E. Obiadoh, O.J Abiola, T.D. Ogunleye, B.A. Tewodros, and T.O Abiola. 2025. Cic-nlp at mahed 2025 task 1:assessing the role of bigram augmentation in multiclass arabic hate and hope speech classification. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

S. S. Poorna and G. J. Nair. 2019. Multistage classification scheme to enhance speech emotion recognition. *International Journal of Speech Technology*, 22(2):327–340.

Md. Abdur Rahman, Md. Sabbir Dewan, Md. Tofael Ahmed Bhuiyan, and Md. Ashiqur Rahman. 2025. Smollab_seu at mahed shared task: Do arabic-native encoders surpass multilingual models in detecting the nuances of hope, hate, and emotion? In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Ibad-ur-Rehman Rashid and Muhammad Hashir Khalil. 2025. Ayahverse at mahed shared task: Fine-tuning arabicbert with preprocessing for hope and hate detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Itbaan Safwan. 2025. Thinking nodes at mahed: A comparative study of multimodal architectures for arabic hateful meme detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Abdallah Saleh and Mariam Biltawi. 2025. Htu at mahed shared task: Ensemble-based classification of arabic hate and hope speech using pre-trained dialectal arabic models. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Octa Syakila Salsabila, Sri Melati Rahmayani, and Isti Yuniastuti. 2024. Content analysis of arabic vocabulary in al quran for the improvement of emotional intelligence. *LiNGUA: Jurnal Ilmu Bahasa dan Sastra*, 19(1):97–108.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

Tran Tran, Trinh and Thin Dang, Van. 2025. Trantranuit at mahed shared task: Multilingual transformer ensemble with advanced data augmentation and optuna-based hyperparameter optimization. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Gerben A. van Kleef and Gert-Jan Lelieveld. 2022. Moving the self and others to do good: The emotional underpinnings of prosocial behavior. *Current Opinion in Psychology*, 44:80–88. Epub 2021-08-31.

Al-Habashi Yasser, Sibaee1 Serry, Nacar Omer, Ammar Adel, and Wadii Boulila1. 2025. Anlpers at mahed shared task: From hate to hope: Boosting arabic text classification. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

Mohamed Zaytoon, Ahmed Salem, Ahmed Sakr, and Hossam Elkordi. 2025. Araminds at mahed 2025: Leveraging vision-language models and contrastive multi-task learning for multimodal hate speech detection. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

8 Appendix

Table 8: Task 1: Text-based Hate/Hope/Emotion Detection

Team	Models	Notable Methods	Compute
NYUAD	GPT-4o-mini, Gemini Flash 2.5 + SVM	Google text embeddings + SVM	OpenAI platform
NguyenTriet	MARBERTv2, AraBERTv0.2-Twitter	Arabic cleanup, ensemble	Tesla P100 (Kaggle)
SmolLab_SEU	MARBERTv2, ARBERTv2, AraBERTv2-large, XLM-R, mDeBERTaV3	Multi-model ensemble	Kaggle P100
AAA	MARBERT, AraBERT-Twitter, XLM-RoBERTa	Arabic tokenization	Tesla V100
KALAM	TF-IDF+LR, AraBERT, CAMeL-BERT, MARBERT	TF-IDF + embeddings + attention	24 GB GPU
AraNLP	AraBERT v0.2-Twitter	AraBERTPreprocessor + 5-fold CV	Google Colab L4
HTU	ArabicDeBERTa-DA, BERT-MSA, MARBERTv2	—	—
REGLAT	AraBERTv2, CAMeL-BERT + SVM/LR	TF-IDF + embeddings, majority voting	Colab GPU
ANLP-UniSo	XLM-RoBERTa, LSTM	SMOTE augmentation	—
trantranuit	AraBERT, XLM-RoBERTa	BoW + TF-IDF + morphological features	Kaggle P100
CIC-NLP	MARBERT	Linguistic + BoW features	RTX 3800, 32 GB RAM
CUET_Zahra_Duo	AraBERTv2-large	Contextual embedding + early stopping	Tesla T4 (32 GB total)
IRIT_HOPE	bert-base-arabertv02-twitter	Token-level augmentation, multi-embedding	—
LoveHeaven	bert-base-arabertv02(-twitter)	Attention-based features	Kaggle P100
AyahVerse	AraBERT	Embeddings + EDA (synonym/back-translation)	—
baoflowin502	AraBERTv2, CAMeL-BERT, BERT Arabic	—	Kaggle P100
Quasar	xlm-roberta-large, gemma-7b, qwen2.5-14b-instruct	Diacritics normalization + synonym balancing	—
TranTranUIT	AraBERTv2, AraBERT-Twitter, XLM-RoBERTa	Dialect sensitivity, cross-lingual + back-translation	—

Table 9: Task 2: Multitask Text Classification

Team	Models	Multitask Setup	Compute
NYUAD	GPT-4o-mini (3 models)	Parallel: separate per sub-task	OpenAI platform
MultiMinds	SVM, XGBoost, AraBERT, GPT-5	Parallel multi-head shared encoder	Colab (6 GB)
NguyenTriet	MARBERTv2, AraBERTv0.2-Twitter	Sequential cascade: Emotion→Offensive→Hate	Kaggle P100
SmolLab_SEU	MARBERTv2, ARBERTv2, XLM-RoBERTa-large	Sequential cascade (3 classifiers)	Kaggle P100
KALAM	CAMeL-BERT, MARBERT, AraBERT	Single-task fine-tuning	24 GB GPU
HTU	Retrospective Reader, ALBERT	—	—
CUET_823	Meta-Llama-3.1-8B	—	Kaggle GPU (16 GB)
Quasar	qwen2.5-14B, gemma-7b, AraBERTv2	—	—

Table 10: Task 3: Multimodal Meme Classification

Team	Models	Fusion / Approach	Compute
NYUAD	GPT-4o-mini, Gemini Flash 2.5, Llama 3.2-11B, Paligemma2	Multimodal embeddings + oversampling	OpenAI + Vertex AI

Team	Models	Fusion / Approach	Compute
thinkingNodes	CLIP-ViT-B/32 + MARBERT	Cross-attention, CNN fusion, contrastive CLIP-Arabic	Kaggle T4 (15 GB)
Araminds	Qwen2.5-1.5B+ResNet / MARBERTv2+ResNet, Gemma3-4B	Dual-encoder + contrastive + VLM ensemble	RTX 3090
MultiMinds	CLIP ViT-B/32 + DistilBERT	ELU-Net cross-attention fusion	Google Colab (6.2 GB)
yassirea	MARBERTv2 + CLIP-Large (ViT-L/14)	4-way fusion + heavy augmentation	RTX 6000 Ada (48 GB)
Muhammad Annas Shaikh	EfficientNet-B0 + AraBERT	—	—
CUET_NLP	mBERT + InceptionResNetV2	—	—
joy_2004114	mBERT, AraBERT, InceptionNet	—	—

NYUAD at MAHED Shared Task: Detecting Hope, Hate, and Emotion in Arabic Textual Speech and Multi-modal Memes Using Large Language Models

Nouar AlDahoul
Computer Science Department
New York University
Abu Dhabi, UAE
nouar.aldahoul@nyu.edu

Yasir Zaki
Computer Science Department
New York University
Abu Dhabi, UAE
yasir.zaki@nyu.edu

Abstract

The rise of social media and online communication platforms has led to the spread of Arabic textual posts and memes as a key form of digital expression. While these contents can be humorous and informative, they are also increasingly being used to spread offensive language and hate speech. Consequently, there is a growing demand for precise analysis of content in Arabic text and meme. This paper explores the potential of large language models to effectively identify hope, hate speech, offensive language, and emotional expressions within such content. We evaluate the performance of base LLMs, fine-tuned LLMs, and pre-trained embedding models . The evaluation is conducted using a dataset of Arabic textual speech and memes proposed in the ArabicNLP MAHED 2025 challenge. The results underscore the capacity of LLMs such as GPT-4o-mini, fine-tuned with Arabic textual speech, and Gemini Flash 2.5, fine-tuned with Arabic memes, to deliver the superior performance. They achieve up to 72.1%, 57.8%, and 79.6% macro F1 scores for task 1, 2, and 3, respectively and secure first place overall in the challenge[1] (Zaghouani et al., 2025). The proposed solutions offer a more nuanced understanding of both text and memes for accurate and efficient Arabic content moderation systems.

1 Introduction

AI content moderation refers to the use of artificial intelligence to monitor, evaluate, and manage content across digital platforms[2]. By ensuring that posts comply with community standards and legal regulations, it helps create safer, more respectful, and law-abiding online environments. Its role has become increasingly vital as the volume and complexity of online content continue to grow. Despite growing efforts, Arabic content moderation still lags behind. Challenges such as dialect diversity, limited training data, and under-resourced tools make it difficult to ensure effective moderation across Arabic-speaking regions[3],[4].

Although Arabic is spoken by around 380 million people, it is far from being a uniform language[5]. It consists of six major regional dialect groups, so for classifiers to work effectively, they must be trained across all these dialects. The rise of social media and online communication platforms has led to the spread of Arabic textual posts and memes as a key form of digital expression. There is a growing need to develop methods for detecting hateful text and memes, as they can perpetuate harmful stereotypes and contribute to the spread of offensive language and hate speech in digital spaces (Zaghouani et al., 2024; Zaghouani and Biswas, 2025a; AlDahoul et al., 2024a).

To have a full understanding of the emotional landscape of online communication, recognition of emotional expression can provide deeper insight into user sentiment and foster empathy. Additionally, emotional expression classification has valuable applications such as monitoring mental health and tailoring personalized recommendations (Zaghouani and Biswas, 2025b).

Memes are especially widespread and can be potent tools for spreading propaganda, inciting hate, or conveying humor. LLMs have been shown to have superior performance in various domains and applications (AlDahoul et al., 2025, 2024b). For meme understanding, having textual and visual inputs, LLMs can analyze both the linguistic content and the underlying visual elements of a meme.

[1] https://marsadlab.github.io/mahed2025/#

[2] https://verpex.com/blog/website-tips/ai-content-moderation

[3] https://techglobalinstitute.com/announcements/blog/content-moderation-arabic-hebrew-in-under-resourced-regions/

[4] https://www.mei.edu/publications/content-moderation-trends-mena-region-censorship-discrimination-design-and-linguistic

[5] https://techglobalinstitute.com/announcements/blog/content-moderation-arabic-hebrew-in-under-resourced-regions/

Our analyses and experiments center around the following research questions: **RQ1**: Can a pre-trained embedding model, combined with trained SVM or DNN classifiers, effectively detect hate and hope speech in Arabic text and memes? **RQ2**: Are existing safety classification and content moderation solutions capable of detecting hate speech in Arabic memes? **RQ3**: To what extent do state-of-the-art base LLMs excel in detecting hate speech in Arabic memes? **RQ4**: Can fine-tuned LLMs detect emotion, hope, hate, and offensive content in Arabic text with high accuracy? **RQ5**: Can fine-tuned LLMs detect hateful Arabic memes with high accuracy?

2 Related Work

Several studies have investigated hate speech and offensive language in Arabic text (Mohaouchane et al., 2019; Kaddoura et al., 2023; Mubarak et al., 2023; Shapiro et al., 2022; Albadi et al., 2018; Bennessir et al., 2022). They utilized Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), CNN-LSTM (Mohaouchane et al., 2019; Kaddoura et al., 2023), word embeddings with simple Recurrent Neural Networks (RNN) (Albadi et al., 2018) and MARBERT (Shapiro et al., 2022; Bennessir et al., 2022). The datasets used for analysis contain social posts and tweets.

To study the proportion of hate speech and offensive language in Arabic tweets, AraBERT was utilized (Zaghouani et al., 2024). They found that 15% of tweets contained offensive language, while 6% included hate speech. Additionally, their annotated tweet dataset provided a valuable contribution to the limited availability of Arabic data related to hate speech and offensive language (Zaghouani et al., 2024). It was found that AraBERT outperformed conventional machine learning classifiers (Zaghouani et al., 2024).

Even though there are several English emotion datasets, there is still a shortage of comprehensive Arabic datasets that support the analysis of both emotions and hope speech. (Zaghouani and Biswas, 2025b) proposed an Arabic dataset, fostering better cross-linguistic analysis of emotions and hope speech. They fine-tuned the AraBERT model (Antoun et al., 2020) for the hate-hope classification task.

Building on previous research, numerous studies broadened the scope to tackle the challenge of detecting Arabic content across multiple modalities. In the context of Arabic propaganda identification (Alam et al., 2024b; Hasanain et al., 2024), separate feature extractors were employed for text and images. Moving from propaganda to hate, a multi-modal analysis of Arabic memes was done to further detect hate in memes. They used a fusion of features extracted from AraBERT for text and ConvNxT for images (Alam et al., 2024a).

3 Materials and Methods

3.1 Dataset Overview

Here we describe the datasets proposed in the ArabicNLP MAHED 2025 challenge (Zaghouani et al., 2025) that we utilized to run our experiments. The **first dataset** is text-based speech that includes 9,843 examples for training, 1,476 for validation, and 1,477 for testing. The goal of using this data is to classify the speech text into one of three categories: hope, hate, or not_applicable.

The **second dataset** is a text-based multi-task set that contains 8,515 examples (5,960 for training, 1,277 for validation, and 1,278 for testing) and supports three types of sub-tasks. The first sub-task aims to classify each text into one of twelve emotions: neutral, anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, or trust. The second sub-task aims to detect offensive language in the text, labeling it as either yes or no. When offensive language is detected, the third sub-task classifies the text as either hate or not_hate.

The **third dataset** targets multi-modal hateful meme detection. It has 4,500 examples (2,143 for training, 312 for validation, and 606 for testing) annotated with two labels: hateful and not hateful. Each meme example includes an image and its extracted Arabic text.

3.2 Methods

Detection of Hope and Hate in Arabic Speech: For this task, first, we fine-tuned 2 LLMs such as GPT-4o-mini[6] (namely LLM 1 in Table 1), and Gemini Flash 2.5[7] (Team et al., 2023)(namely LLM 2 in Table 1) using the training and validation sets from the **first dataset**. Secondly, we utilized Google text embedding[8]+ SVM (Hearst et al.,

[6] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[7] https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking

[8] https://developers.googleblog.com/en/gemini-embedding-text-model-now-available-gemini-api

1998) (namely LLM 3 in Table 1). To improve the accuracy, we used ensemble learning (namely Ensemble in Table 1) that used majority voting among previous 3 models. We found that many hope samples were predicted as not_applicable. So we added hope/not_applicable fine-tuned GPT-4o-mini to address this issue. we named our solution in Table 1. We reported the results of inference on a testing set.

Multi-task Detection of Emotional Expressions, Offensive Language, and Hate Speech: For this task, three GPT-4o-mini models were fine-tuned using the training and validation sets from the **second dataset** for three epochs with a learning rate multiplier of 1.8. We reported the results of inference on a testing set.

To address the class imbalance in hate/not-hate sub-task, we over-sampled the minority 'hate' text by a factor of five to achieve a more balanced distribution between the 'hate' and 'not-hate' classes.

Multi-modal Detection of Arabic Hateful Memes: For this task, we have evaluated several methods, including base LLMs, fine-tuned LLMs, and embedding models, to find the best solution. We tested all solutions using the testing data of 606 Arabic memes.

First, we started with assessing the performance of **embedding models** that can combine their outputs with traditional classifiers for hate/not-hate classification. We used the Google multi-modal pre-trained embedding model (multimodalembedding@001)[9] to generate embedding vectors for each text and image in each meme. The embedding vector has 512 dimensions. Later, we aggregated the two embedding vectors of text and image by computing their element-wise average first and then by concatenating the two vectors. Finally, we added a support vector machine (SVM) (Hearst et al., 1998) to classify the resulting embedding vector into two classes: hate and not-hate. We assessed four scenarios: text embedding vector only, image embedding vector only, average of text and image embedding vectors, and concatenation of text and image embeddings. We fine-tuned hyperparameters of SVM to get the highest F1 and F2 scores. We found that regularization parameter C = 0.1, kernel = radial basis function (rbf), gamma = scale, and balanced class weighted loss function are the optimal hyperparameters for the three scenarios except the text-only scenario, where C = 1 is optimal. Additionally, we replaced SVM with a deep neural network (DNN) (LeCun et al., 2015) whose architecture was optimized to get the optimal one with the highest F1 and F2 scores.

In the **second experiment**, we assessed the capacity of multi-modal pre-trained **safety classifiers** for hate detection in memes.

Llama Guard 4[10],[11] (Chi et al., 2024) is a multi-modal safety classifier with 12 billion parameters, trained jointly on both text and images. It uses a dense architecture derived from the Llama 4 Scout pre-trained model, which has been pruned and fine-tuned specifically for content safety classification. In this work, our focus is on the 'hate' category, which refers to text that demeans or dehumanizes individuals based on sensitive personal characteristics. We focus on all examples that have been flagged under the hate category only.

Omni-moderation-latest[12] is a moderation endpoint used to check whether text or images are potentially harmful. Its output includes several categories and their confidence values. The moderator sets the flag to true if it classifies the content as harmful. The limitation of this moderator is that for categories such as 'hate' or 'hate/threatening,' it supports only text. We consider all examples that have triggered the safety flag.

In the **third experiment**, we ran **Gemini Flash 2.5**, a base model with a system prompt (Prompt 1). We also ran the **GPT-4o-mini** base model with Prompts 1, 2, and 3 (available in the Appendix).

To improve the detection performance, we fine-tuned several LLMs in a supervised learning setting. We started by tuning Gemini Flash 2.5 using Prompt 3. To address the class imbalance, we over-sampled the minority 'hate' memes by a factor of nine to achieve a more balanced distribution between the 'hate' and 'no_hate' classes. The hyper-parameters used are three epochs, learning_rate_multiplier of 0.5, an adapter size of 2, an off threshold in safety_settings, and disabled thinking. Additionally, we also fine-tuned **Llama 3.2-11B**[13] (Dubey et al., 2024) using both text and image inputs from the training data. We used

/

[9] https://cloud.google.com/vertex-ai/generative-ai/docs/embeddings/get-multimodal-embeddings

[10] https://www.llama.com/docs/model-cards-and-prompt-formats/llama-guard-4/

[11] https://huggingface.co/meta-llama/Llama-Guard-4-12B

[12] https://platform.openai.com/docs/guides/moderation

[13] https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct

Low-Rank Adaptation (LoRA) (Hu et al., 2022) as the Parameter-Efficient Fine-Tuning (PEFT) (Xu et al., 2023) method for fine-tuning utilizing the unsloth framework. The fine-tuned Llama 3.2-11B model was uploaded to Huggingface: https://huggingface.co/NYUAD-ComNets/Llama3.2-MultiModal-Hate_Detector_Memes

Finally, we fine-tuned **Paligemma2**[14] (Steiner et al., 2024) namely "google/paligemma2-3b-pt-224". The parameters of the vision tower and the language model are frozen, while only the parameters of the multi-modal projector are set to be trainable.

In the previous fine-tuning experiments, we used OpenAI[15] for tuning each GPT-4o-mini. Additionally, we used Google AI vertex studio[16] for tuning Gemini Flash 2.5.

4 Results and Discussion

Hate/Hope Detection in textual speech: In this task, the ensemble method of majority voting among the three LLMs improved the performance as shown in Table 1. Moreover, adding the hope/not classifier to better distinguish real hope samples from those predicted as not_applicable achieved the best performance metrics and ranked second in the leaderboard (Zaghouani et al., 2025) which addresses **RQ4**. It is also worth mentioning that embedding model + SVM (LLM3) shows good performance which answers **RQ1**.

Task	Accuracy %	Macro Precision %	Macro Recall %	Macro F1 Score %
LLM 1	70.6	70.6	69	69.7
LLM 2	69.7	68.6	72.2	69.9
LLM 3	70.6	71.6	67.2	68.9
Ensemble	71.9	71.7	71.2	71.4
Our Solution	**72.3**	71.6	**72.9**	**72.1**

Table 1: Performance metrics for Task 1 (hop/hate/not_applicable)

Multi-task Detection: The three fine-tuned GPT-4o-mini for multi-task (emotion, offensive, hate) achieved the best performance compared to other methods in the leaderboard (Zaghouani et al., 2025) evaluated on a testing set which addresses **RQ4**. More details in Table 2. The model achieved a Macro F1-score of 57.8%, an accuracy of 75.0%, a precision of 61.2%, and a recall of 57.8% over all three sub-tasks.

Task	Accuracy %	Macro Precision %	Macro Recall %	Macro F1 score %
Emotion	59.9	57.2	49.9	51.7
Offensive/ Not	85.4	82.0	84.8	83.1
Hate/Not	63.8	-	-	-

Table 2: performance metrics for multi-task (task 2). Hate/Not detection is influenced by the offensive detection step, and some evaluation metrics cannot be computed because samples predicted as non-offensive yield NaN values and are excluded from the Hate/Not detector.

Hate Detection in Memes: Table 3 presents performance metrics for a variety of models. A pre-trained multi-modal embedding model was found to effectively detect hate speech in Arabic memes using either SVM or DNN, answering **RQ1**. Both LLaMA 4 Guard and OpenAI content moderator show lower recall and F1-scores, especially OpenAI one, suggesting limitations in the existing safety classification solutions on this task, which addresses **RQ2**. Among the base LLMs, GPT-4o demonstrated stronger performance compared to Gemini Flash 2.5, answering **RQ3**.

Fine-tuned Gemini Flash 2.5 demonstrates superior performance across all metrics. Similarly, fine-tuned Llama 3.2 11B consistently ranks second. The results indicate that fine-tuning significantly boosts models' capabilities, which addresses **RQ5**. On the other hand, fine-tuned PaliGemma2 underperforms compared to other models.

Table 4 shows Google's multi-modal embedding model results with SVM for different input modalities. The findings indicate that the average embedding vector outperforms slightly the image-only embedding. This suggests that adding text embeddings does not provide an advantage for classification. One explanation is that Google's embedding model processes the text within the meme's image. The performance of text-only embeddings is the lowests. We also ran GPT-4o-mini with the three prompts as shown in Table 5. Even though Prompt 3 produced the highest accuracy and macro F1 score, Prompt 1 gave the highest macro F2 score, suggesting a better prompt to detect the hate class specifically.

Flash Flash 2.5 achieved the best performance in

[14] https://huggingface.co/google/paligemma-3b-pt-224

[15] https://platform.openai.com/finetune/

[16] https://console.cloud.google.com/vertex-ai/studio/

LLM	Accuracy %	Macro Precision %	Macro Recall %	Macro F1 score %	Macro F2 score %
embedding + SVM	77.56	70.48	70.83	70.65	70.76
embedding + DNN	77.56	70.32	69.97	70.14	70.04
OpenAI content moderator	72.77	57.27	52.85	51.18	51.75
Llama 4 Guard	71.45	63.36	64.38	63.77	64.11
GPT-4o-mini	79.21	72.49	71.29	71.84	71.50
Gemini Flash 2.5	64.19	62.47	66.36	61.09	63.20
Fine-tuned Gemini Flash 2.5	**83.33**	**78.84**	**74.91**	**76.49**	**75.46**
Fine-tuned Llama 3.2 11B	80.36	74.09	73.14	73.58	73.31
Fine-tuned Paligemma2	76.73	68.95	67.49	68.12	67.72

Table 3: Performance of base and fine-tuned LLMs for task 3.

the leaderboard (Zaghouani et al., 2025) evaluated on a testing set of 500 memes. The model achieved a Macro F1-score of 79.6%, an accuracy of 80.0%, a precision of 79.4%, and a recall of 80.4%.

Limitations

One limitation of this work is the subjective nature of the annotations poses challenges, as different annotators may interpret and label content differently. This introduces potential inconsistencies in the training data, which could affect the model's performance.

Another key limitation is the models' ability to understand and process different Arabic dialects.

	Accuracy %	Macro Precision %	Macro Recall %	Macro F1 score %	Macro F2 score %
image embedding + SVM	77.23	70.10	70.61	70.34	70.50
text embedding + SVM	66.50	57.15	57.64	57.33	57.50
Avg. embedding of image&text + SVM	**77.56**	**70.48**	**70.83**	**70.65**	**70.76**
Concatenate embedding of image&text + SVM	76.07	68.53	68.76	68.64	68.71

Table 4: Performance of different input modalities combined with evaluated on validation set in task 3.

	Accuracy %	Macro Precision %	Macro Recall %	Macro F1 score %	Macro F2 score %
GPT-4o-mini Prompt 1	74.75	70.70	**76.23**	71.34	**73.62**
GPT-4o-mini Prompt 2	79.21	72.49	71.29	71.84	71.50
GPT-4o-mini Prompt 3	**82.51**	**80.86**	69.44	**72.29**	70.14

Table 5: GPT-4o-mini base model performance under different prompts evaluated on validation set in task 3.

References

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmed, Md Arid Hasan, and Maram Hasanain. 2024b. Armeme: Propagandistic content in arabic memes. *arXiv preprint arXiv:2406.03916*.

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76. IEEE.

Nouar AlDahoul, Myles Joshua Toledo Tan, Harishwar Reddy Kasireddy, and Yasir Zaki. 2024a. Advancing content moderation: Evaluating large language models for detecting sensitive content across text, images, and videos. *arXiv preprint arXiv:2411.17123*.

Nouar AlDahoul, Myles Joshua Toledo Tan, Harishwar Reddy Kasireddy, and Yasir Zaki. 2024b. Exploring vision language models for facial attribute recognition: Emotion, race, gender, and age. *arXiv preprint arXiv:2410.24148*.

Nouar AlDahoul, Myles Joshua Toledo Tan, Raghava Reddy Tera, Hezerul Abdul Karim, Chee How Lim, Manish Kumar Mishra, and Yasir Zaki. 2025. Multitasking vision language models for vehicle plate recognition with vehiclepaligemma. *Scientific Reports*, 15(1):1–15.

Wissam Antoun, Fady Baly, and Hazem Hajj.

2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Mohamed Aziz Bennessir, Malek Rhouma, Hatem Haddad, and Chayma Fourati. 2022. icompass at arabic hate speech 2022: Detect hate speech using qrnn and transformers. In *Proceedinsg 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 176–180.

Jianfeng Chi, Ujjwal Karn, Hongyuan Zhan, Eric Smith, Javier Rando, Yiming Zhang, Kate Plawiak, Zacharie Delpierre Coudert, Kartikeya Upasani, and Mahesh Pasupuleti. 2024. Llama guard 3 vision: Safeguarding human-ai image understanding conversations.

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. *arXiv e-prints*, pages arXiv–2407.

Maram Hasanain, Md Arid Hasan, Fatema Ahmed, Reem Suwaileh, Md Rafiul Biswas, Wajdi Zaghouani, and Firoj Alam. 2024. Araieval shared task: propagandistic techniques detection in unimodal and multimodal arabic content. *arXiv preprint arXiv:2407.04247*.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Sanaa Kaddoura, Suja A Alex, Maher Itani, Safaa Henno, Asma AlNashash, and D Jude Hemanth. 2023. Arabic spam tweets classification using deep learning. *Neural Computing and Applications*, 35(23):17233–17246.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

Hanane Mohaouchane, Asmaa Mourhir, and Nikola S Nikolov. 2019. Detecting offensive language on arabic social media using deep learning. In *2019 sixth international conference on social networks analysis, management and security (SNAMS)*, pages 466–471. IEEE.

Hamdy Mubarak, Sabit Hassan, and Shammur Absar Chowdhury. 2023. Emojis as anchors to detect arabic offensive language and hate speech. *Natural Language Engineering*, 29(6):1436–1457.

Ahmad Shapiro, Ayman Khalafallah, and Marwan Torki. 2022. Alexu-aic at arabic hate speech 2022: Contrast to classify. *arXiv preprint arXiv:2207.08557*.

Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, et al. 2024. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*.

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*.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

A Appendix

A.1 Prompts used

The following prompts were used in this work for the three tasks. The prompt in orange color aims to classify text into hate, hope, and not_applicable. The three prompts in cyan color aim to classify text into:

- offensive language or not.
- hate or not.
- one of the emotions in a predefined set of emotions.

The prompts in red, blue, and green color are three different prompts used to classify memes into hate or not.

Hate/Hope Detection Prompt

Classify each text into one of the three categories ’hope’,’hate’,or ’not_applicable’

Offensive Language Detection Prompt

You are an expert in offensive language detection.
Carefully read the input text and determine whether it is offensive.
Respond with only one word, either "yes" if the text is offensive, or "no" if it is not.
Text: "<your text here>"
Answer:

Hate/not-Hate Detection Prompt

Classify each text into one of the two categories ’hate’, or ’not_hate’

Emotion Detection Prompt

You are an expert in fine-grained emotion classification.
Given a short piece of text, your task is to classify the **dominant emotion** expressed in the text.
Use **only one word** from the following list of predefined emotion categories:

[anger, disgust, neutral, love, joy, anticipation, optimism, sadness, confidence, pessimism, surprise, fear]
Carefully read the input and select the most appropriate emotion based on the **overall tone, context, and word choice**.
If multiple emotions are expressed, choose the one that is **most central** to the message.
Respond with only one word, exactly as listed.
Text: "<your text here>"
Answer:

Hate Detection in Memes Prompt 1

You’re an expert Arabic social media hate content analyst. Your goal is to flag any content that is derogatory, disparaging, promotes negative stereotypes, incites any form of harm (even humorously), or contributes to a hostile environment.
Provide the following for analysis:
Meme Description (Visuals): Detailed text description of all visual elements (characters, symbols, actions, etc.).
Meme Text (Arabic & English): Exact Arabic text and its literal English translation. Include any emojis/symbols.
Context: Explain who is targeted (individuals, groups, professions, etc.), relevant current events, cultural/historical references, and any known slang/derogatory terms.
User Intent (if known): Any information on the creator/sharer’s likely purpose.
Analysis: Your detailed reasoning, explaining how visuals, text, and context contribute to hate.
Focus on dehumanization, negative stereotypes, incitement (even if satirical), hostile environment creation, contempt, ridicule, disgust, targeting based on role, disparaging language (ableism, body shaming, etc.), and normalization of problematic behavior.
Final Answer: hate/no hate

Hate Detection in Memes Prompt 2

You are a hate speech analyst evaluating Arabic social media memes.
Your goal is to flag hate speech, including implicit hate masked as humor or political discourse.
Hate includes:
Stereotyping or mocking a group based on ethnicity, nationality, gender, religion, or identity Jokes or memes that promote harmful generalizations
Political content that blames or dehumanizes a people, not just a government or policy
However, you must not flag:
Political critique focused on leadership, government, or policy (not people)
Cultural observations or satire without negative generalization
Your analysis must:
Describe the image and interpret the Arabic text
Explain whether it includes group-based bias or stereotypes
End with:
Analysis: your reasoning
Final Answer: ’hate’ or ’no hate’

Hate Detection in Memes Prompt 3

You are a hate speech analyst evaluating Arabic social media memes.
Your goal is to classify meme into hate or no_hate

A.2 Hyper-parameters for various models

The following are Hyper-parameters used for training DNN, and fine-tuning PaliGemma2, and Llama 3.2- 11B. Table 6 describes the DNN’s architecture.

Hyper-parameters for DNN

- Adam optimizer,
- weighted class binary cross-entropy loss fuction
- 100 epochs
- 128 batch size
- early stopping with patience = 3.

Training Configuration of PaliGemma2

- number of training epochs: 3
- per-device training batch size: 2
- gradient accumulation steps: 8
- warm-up steps: 2
- learning rate: 2e-5
- weight decay: 1e-6
- Adam optimizer beta2 value: 0.999
- optimizer type: Adamw_hf
- early stopping callback with patience=2.

Fine-tuning Configurations of Llama 3.2-11B

- the training batch size per device is set to 4.
- gradients are accumulated over 4 steps.
- the learning rate warm-up lasts for 5 steps.
- the total number of training steps is 150.
- the learning rate is set to 0.0002.
- the optimizer used is 8-bit AdamW
- weight decay is set to 0.01.
- a linear learning rate scheduler is used.

Layer Type	Output Shape	Activation	Description
Input Layer	(512,)	–	Input vector representing image embedding
Dense Layer	(256,)	ReLU	Fully connected layer on image input
Dropout	(256,)	Dropout 0.5	Regularization
Dense Layer	(128,)	ReLU	Further transformation of image embedding
Dropout	(128,)	Dropout 0.5	Regularization
Dense Layer	(64,)	ReLU	Compressed feature representation
Dropout	(64,)	Dropout 0.5	Regularization
Input Layer	(512,)	–	Input vector representing text embedding
Dense Layer	(256,)	ReLU	Fully connected layer on text input
Dropout	(256,)	Dropout 0.5	Regularization
Dense Layer	(128,)	ReLU	Intermediate transformation
Dropout	(128,)	Dropout 0.5	Regularization
Dense Layer	(64,)	ReLU	Compressed feature representation
Dropout	(64,)	Dropout 0.5	Regularization
Concatenate	(128,)	–	Merge image and text features (64 + 64)
Dense Layer	(128,)	ReLU	Combined representation processing
Dropout	(128,)	Dropout 0.5	Regularization
Dense Layer	(1024,)	ReLU	High-capacity layer for rich interaction
Dropout	(1024,)	Dropout 0.5	Regularization
Dense Layer	(1,)	Sigmoid	Final prediction for binary classification

Table 6: Architecture of the dual-branch DNN model for image and text fusion.

A.3 Confusion matrices for various models

The following are confusion matrices presenting the models' performance in terms of False Positives, False Negatives, True Positives, and True Negatives.

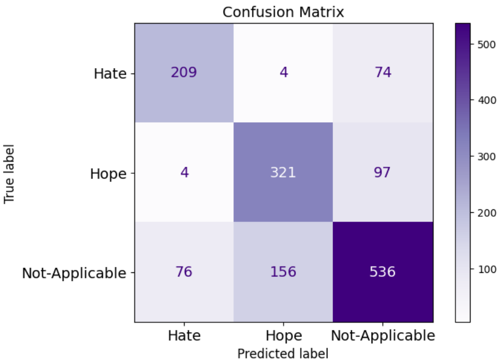


Figure 1: Confusion matrix of testing set in task 1 for hope/hate/not_applicable classification in text using ensemble of 3 fine-tuned LLMs (gpt-4o-mini, Gemini Flash 2.5, and Google text embedding + SVM) + fine-tuned gpt-4o-mini for hope/not

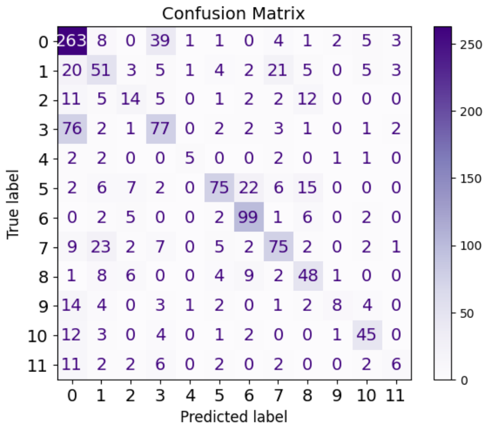


Figure 2: Confusion matrix of testing set in task 2 for emotion classification in text using Fine-tuned GPT-4o-mini. class 0: Anger, class 1: Anticipation, class 2: Confidence, class 3: Disgust, class 4: Fear, class 5: Joy, class 6: Love, class 7: Neutral, class 8: Optimism, class 9: Pessimism, class 10: Sadness , class 11: Surprise

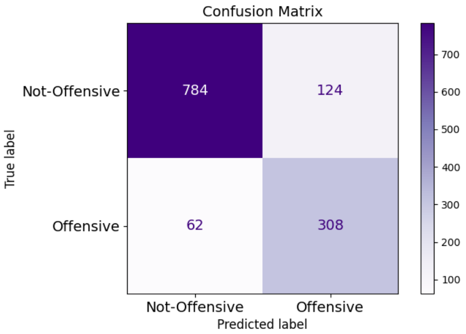


Figure 3: Confusion matrix of testing set in task 2 for offensive detection in text using Fine-tuned GPT-4o-mini.

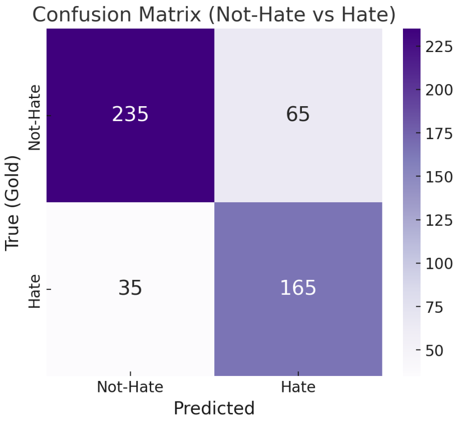


Figure 4: Confusion matrix of testing set for hate detection in memes using Fine-tuned Gemini Flash 2.5

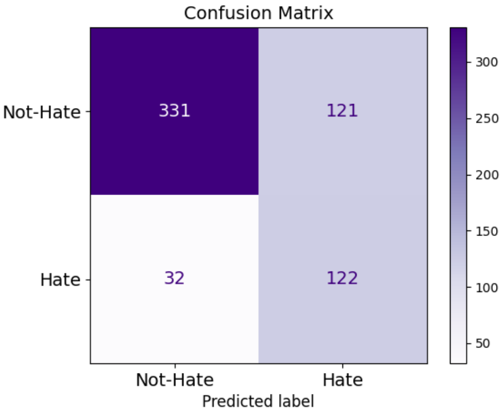


Figure 5: Confusion matrix of validation set for hate detection in memes using GPT-4o-mini with Prompt 1

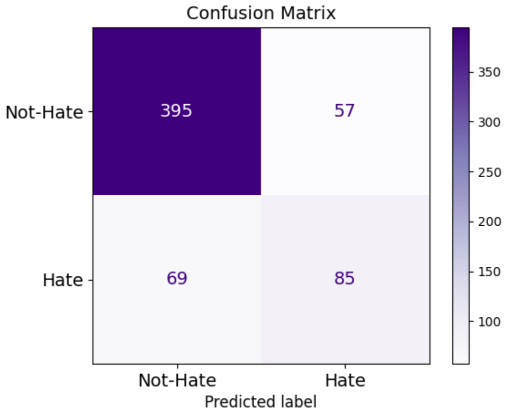


Figure 6: Confusion matrix of validation set for hate detection in memes using GPT-4o-mini with Prompt 2

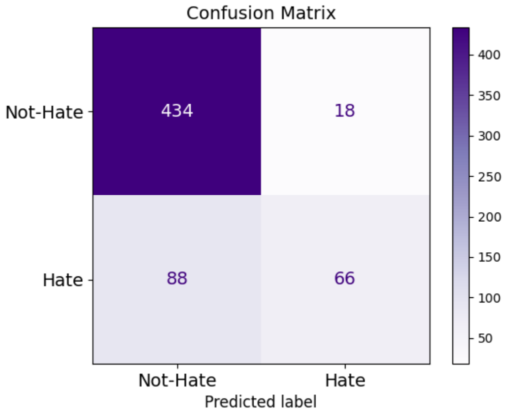


Figure 7: Confusion matrix of validation set for hate detection in memes using GPT-4o-mini with Prompt 3

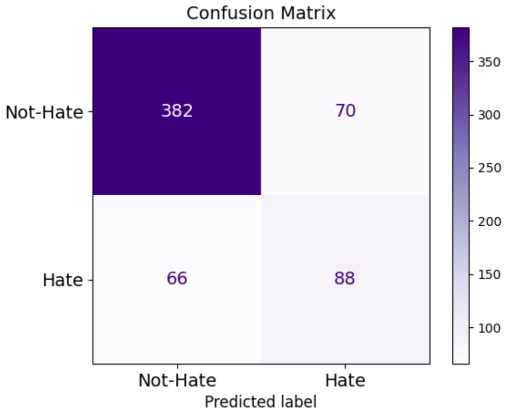


Figure 8: Confusion matrix of validation set for hate detection in memes using average embeddings of image and text + SVM

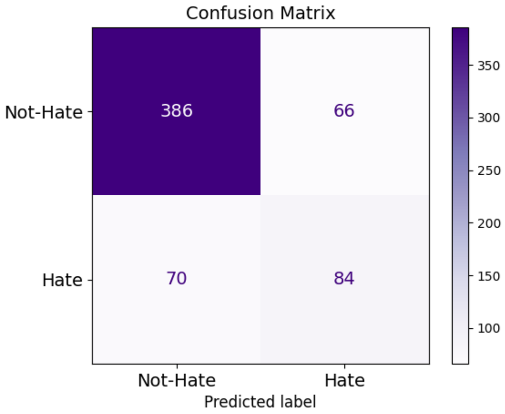


Figure 9: Confusion matrix of validation set for hate detection in memes using average embeddings of image and text + DNN

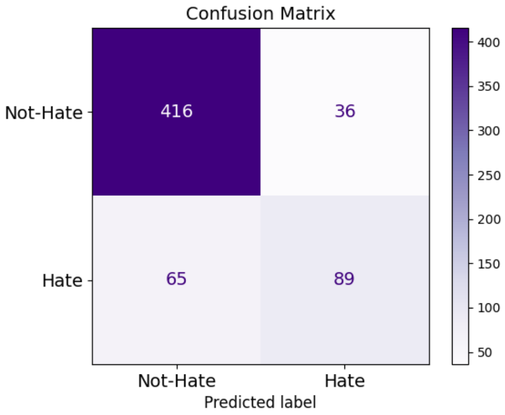


Figure 10: Confusion matrix of validation set for hate detection in memes using Fine-tuned Gemini Flash 2.5

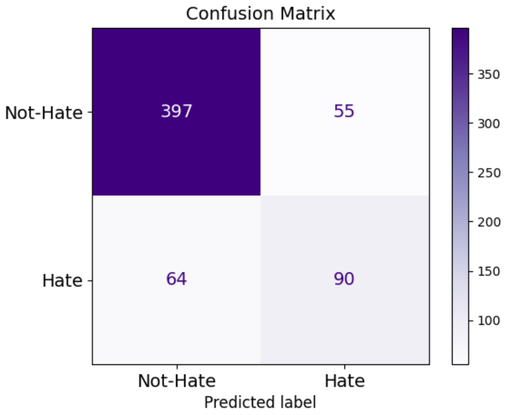


Figure 11: Confusion matrix of validation set for hate detection in memes using Fine-tuned Llama 3.2 11B

NguyenTriet at MAHED Shared Task: Ensemble of Arabic BERT Models with Hierarchical Prediction and Soft Voting for Text-Based Hope and Hate Detection

Nguyen Minh Triet and **Dang Van Thin**
University of Information Technology-VNUHCM
Quarter 34, Linh Xuan Ward, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
23521652@gm.uit.edu.vn and thindv@uit.edu.vn

Abstract

We present the NguyenTriet system for the MAHED 2025 shared task on multimodal detection of hope and hate emotions in Arabic content (Zaghouani et al., 2025). The challenge was divided into three subtasks: text-based hate and hope speech classification in Arabic text; multitask emotion, offensive language, and hate detection in Arabic text with a hierarchical structure; and detecting hateful memes from multimodal text-image pairs. Our participation focused on Subtasks 1 and 2. For Subtask 1, we employed an ensemble of Arabic BERT models for multi-class classification. In Subtask 2, we implemented a hierarchical classification framework utilizing a similar ensemble methodology, where emotion predictions are leveraged through a cascaded pipeline architecture to inform downstream hate and offensive detection tasks. Our approach achieved macro-F_1 scores of 0.707 (3rd place) on Subtask 1 and 0.553 (2nd place) on Subtask 2.

1 Introduction

The detection of hope and hate emotions in multimodal Arabic content has become increasingly critical in the era of social media, where memes and text-based posts can rapidly disseminate polarizing messages (Zaghouani et al., 2024a; Alam et al., 2024b). The MAHED 2025 Shared Task addresses this challenge through three subtasks: (1) text-based hate and hope speech classification in Arabic text, (2) multitask emotion, offensive, and hate detection in Arabic text with a hierarchical structure encompassing emotion classification, offensiveness detection, and hate speech identification, and (3) multimodal hateful meme detection combining Arabic text and images. This task is particularly important for Arabic, a language with diverse dialects and cultural nuances, where automated detection can aid in moderating harmful content while promoting positive discourse (Zaghouani et al., 2025).

Our system employs transformer-based models fine-tuned on the provided datasets, leveraging ensemble techniques and emotion-aware inputs to handle the hierarchical nature of the subtasks. For Subtask 1, we focus on multi-class classification of memes into hate, hope, or not_applicable categories using soft voting ensembles of Arabic-specific BERT variants. For Subtask 2, we adopt a cascaded pipeline that first predicts emotions, then incorporates these predictions into offensiveness and hate detection models. Key findings include the effectiveness of emotion integration in improving downstream tasks and the robustness of ensembles in handling class imbalances. Experiments demonstrate that our ensemble approach enhances performance on imbalanced datasets, with final scores of 0.707 (ranking 3rd) on Subtask 1 and 0.553 (ranking 2nd) on Subtask 2. Our approach achieved competitive rankings, highlighting challenges such as label imbalance, dialectal variations, disambiguating subtle emotions like pessimism due to limited examples, and dialectal ambiguity.

2 Background

2.1 Data

The dataset of MAHED 2025 includes Modern Standard Arabic (MSA) and various dialects, with genres primarily from social media content such as tweets and memes. All content was collected from public social media, anonymized, and annotated by native speakers. The task setup involves three subtasks:

Subtask 1 (Text-based Hate and Hope Speech Classification): Classifying Arabic text into three categories: ’hate’ (content propagating hostility or prejudice), ’hope’ (content inspiring positivity or optimism), or ’not_applicable’ (neutral or unrelated content). The dataset (Zaghouani et al., 2024b; Zaghouani and Biswas, 2025b) contains 9,843 instances with notable class imbalance:

Proceedings of The Third Arabic Natural Language Processing Conference, pages 585–589
November 8-9, 2025 ©2025 Association for Computational Linguistics

’not_applicable’ dominates at 53.36%, followed by ’hope’ (27.65%) and ’hate’ (18.97%).

Subtask 2 (Emotion, Offensive, and Hate Detection - Multitask): Hierarchical classification framework with three sequential stages: (1) emotion classification among 12 categories (neutral, anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust), (2) binary offensiveness detection, and (3) conditional hate classification applied only to offensive content. The dataset (Zaghouani et al., 2024b; Zaghouani and Biswas, 2025a) comprises 8,515 instances with significant imbalances across all levels: ’anger’ dominates emotions (25.94%) while ’fear’ represents only 0.91%; offensiveness skews toward ’no’ (70.79%); hate labels favor ’not_hate’ (82.40% among offensive samples).

Subtask 3 (Multimodal Hateful Meme Detection): Binary classification of Arabic memes requiring analysis of both visual content and embedded Arabic text to determine ’hateful’ or ’non-hateful’ labels. The dataset (Alam et al., 2024b) contains 3,561 instances with class imbalance favoring ’non-hateful’ memes (75.31%).

2.2 Related Work

Related work encompasses several key efforts in Arabic NLP for hate speech, emotion detection, and multimodal analysis.

Prior studies have explored multi-label classification of hate speech from social media tweets and focused analyses of harmful content, providing baselines for binary or multi-class detection (Zaghouani et al., 2024a; Biswas and Zaghouani, 2025a).

Bilingual approaches to emotions and hope speech have advanced positive discourse identification through paired language modeling (Biswas and Zaghouani, 2025b).

In the multimodal domain, investigations into propagandistic content in Arabic memes have established baselines for detecting harmful visual-textual combinations (Alam et al., 2024b), with extensions employing multi-agent large language models for nuanced propaganda analysis (Alam et al., 2024a).

Furthermore, propaganda span annotation has utilized large language models for fine-grained identification in news articles and memes (Hasanain et al., 2024a,b), demonstrating the efficacy of LLMs in capturing subtle spans while often neglecting hierarchical emotion integration.

Participating in subtask 1 and 2, our contribution’s novelty lies in combining soft-voting ensembles of Arabic-specific BERT models with a cascaded emotion-integrated pipeline for hierarchical detection. This approach enhances robustness against class imbalances and dialectal variations, outperforming prior single-model methods or non-cascaded ensembles by explicitly leveraging predicted emotions to inform offensiveness and hate predictions in a structured manner.

3 System Overview

3.1 Approach

Our system comprises key components, including text preprocessing, classifiers formed through soft voting ensembles of Arabic-specific BERT models, and a hierarchical structure designed for Subtask 2 to address the task’s inherent hierarchical nature.

3.2 Text Preprocessing

A critical component of our system is the text preprocessing pipeline, which addresses challenges such as dialectal variations, noisy social media content (e.g., emojis, URLs, and mentions), and orthographic inconsistencies in Arabic script. The preprocessing function is implemented as follows:

- Demojize emojis to their Arabic descriptions using the `emoji` library.
- Strip tashkeel (diacritics), tatweel (elongation), and normalize ligatures with `pyarabic.araby`.
- Normalize alef maksura and teh marbuta using `camel_tools.utils.normalize`.
- Remove URLs, mentions, hashtags, and non-alphanumeric characters (except punctuation like !?.) via regular expressions.
- Remove Arabic stopwords from NLTK’s Arabic stopwords list.

This pipeline reduces text length and noise, improving model focus on semantic content.

3.3 Pre-trained Models

We employed two Arabic-specific BERT models, both pre-trained on extensive Arabic social media corpora, to capitalize on their robust understanding of dialectal variations and informal language patterns characteristic of tweets and social media content:

- **MARBERTv2** (Abdul-Mageed et al., 2021): A comprehensive model designed to handle both Dialectal Arabic (DA) and Modern Standard Arabic (MSA). Pre-trained using masked language modeling (MLM) on a substantial corpus of approximately 1 billion Arabic tweets, this model demonstrates exceptional performance on social media-related NLP tasks across diverse Arabic linguistic varieties and regional dialects.

- **AraBERTv0.2-Twitter** (Antoun et al.): A specialized variant optimized specifically for Arabic dialectal content and Twitter-style communications, built upon the BERT-Base architecture. Through continued pre-training via MLM on approximately 60 million curated Arabic tweets, this model incorporates an extensive vocabulary of dialectal expressions and colloquialisms, making it exceptionally well-suited for processing noisy, abbreviated social media text with informal linguistic structures.

3.4 Systems Details

The systems for Subtasks 1 and 2 are built upon classifiers structured as follows.

Subtask 1: The architecture consists of a single multi-class classifier that receives processed text and performs classification into three labels: hate, hope, or not_applicable.

Subtask 2: The architecture employs a hierarchical structure comprising three classifiers: (1) an emotion classifier for 12 emotion categories, (2) a binary offensiveness classifier that incorporates the predicted emotion as additional context, and (3) a binary hate classifier applied only to samples predicted as offensive, similarly augmented with the emotion label. In the training phase, the three classifiers are trained sequentially: first the emotion classifier, followed by the offensiveness classifier, and finally the hate classifier. In this process, the predicted emotion from the emotion classifier is replaced with the true emotion label to ensure accurate context augmentation for the downstream offensiveness and hate classifiers.

A classifier comprises two models (MARBERTv2 and AraBERTv0.2-Twitter) that perform tokenization and prediction independently. We applied the simple soft voting technique to merge the predictions of the two models, in which we sum up the probability output of the two classifiers and choose the sentiment class with the highest probability as the final prediction.

The hierarchical architecture for Subtask 2 and a classifier architecture are illustrated in Figure 1.

4 Experimental Setup

Data split usage: We utilized the provided train, validation, and test sets for both subtasks. The training set was used exclusively for model training, the validation set for hyperparameter tuning and evaluation during development, and the test set for final evaluation. No data augmentation or splitting was applied beyond the provided sets.

Configuration Settings: All experiments were conducted using a P100 GPU on the Kaggle platform. For both subtasks, the hyperparameters selected to train the two models included a learning rate of 3e-5, weight decay of 0.1, batch size of 32 for MARBERTv2 and 16 for arabert-twitter, over 2 epochs. The loss function employed was a class-weighted CrossEntropyLoss to effectively handle class imbalances during training. Optimization was performed using AdamW with a cosine annealing learning rate scheduler.

Evaluation Metrics: Task evaluation metrics are summarized as macro-averaged F1-score for all subtasks, as per the official guidelines, emphasizing balanced performance across imbalanced classes.

External Tools and Libraries: transformers (v4.20.0), torch (v2.0.0), pandas (v2.0.0), numpy (v1.24.0), scikit-learn (v1.2.0), pyarabic (v0.6.14), emoji (v2.0.0), camel_tools (v1.2.0), nltk (v3.8.0), and scipy (v1.10.0).

5 Results

5.1 Official Results

The official evaluation was conducted on the test set using macro-averaged F1-score. Our ensemble system achieved a macro-F1 of 0.707, ranking 3rd on Subtask 1. For Subtask 2, the system obtained a macro-F1 of 0.553, ranking 2nd. These results represent the official submission scores. The top 3 teams' results in subtask 1 and 2 are demonstrated in Table 1 and 2.

Ranking	Team	Macro-F1
Top 1	HTU	0.723
Top 2	NYUAD	0.721
Top 3 (Ours)	NguyenTriet	0.707

Table 1: Top 3 rankings for Subtask 1 on the test set.

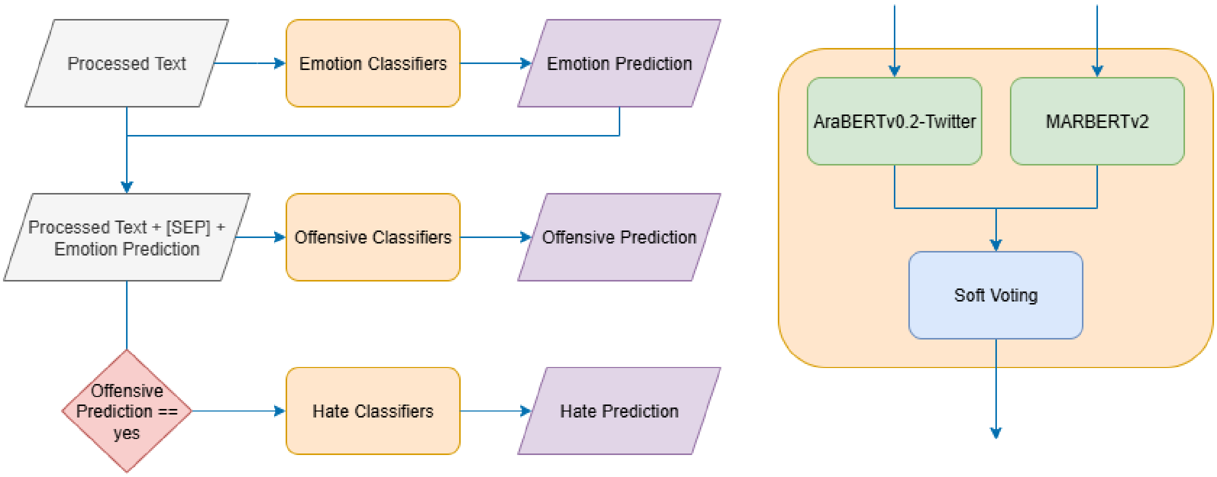


Figure 1: Hierarchical architecture for Subtask 2 (left) and a classifier architecture (right).

Ranking	Team	Macro-F1
Top 1	NYUAD	0.578
Top 2 (Ours)	NguyenTriet	0.553
Top 3	HTU	0.535

Table 2: Top 3 rankings for Subtask 2 on the test set.

5.2 Analysis

We first compare performance across different settings on the test set for Subtask 1, including individual models (MARBERTv2, AraBERTv0.2-Twitter) and the ensemble setting (combining MARBERTv2, AraBERTv0.2-Twitter using soft-voting). Table 3 summarizes the performance for Subtask 1.

Configuration	Macro-F1
MARBERTv2	0.692
AraBERTv0.2-Twitter	0.698
Ensemble	0.707

Table 3: Performance comparison for Subtask 1 across configurations on test set.

Next, for Subtask 2, we compare settings on the test set, distinguishing multiclass (non-hierarchical) and hierarchical configurations for individual models (multiclass MARBERTv2, multiclass arabert-twitter-large, hierarchical MARBERTv2, hierarchical arabert-twitter-large) and ensembles (multiclass Ensemble, hierarchical Ensemble). Table 4 summarizes the performance for Subtask 2.

These comparisons demonstrate the effectiveness of the ensemble architecture, which consistently outperforms individual models by 1-2% across both subtasks on test set, highlighting its role in enhancing robustness and reducing variance. Additionally, the hierarchical (cascaded) structure in Subtask 2 proves superior to multiclass approaches, improving macro-F1 by 4-5%, as it better captures dependencies between emotion, offensiveness, and hate predictions through contextual augmentation.

Configuration	Macro-F1
Multiclass MARBERTv2	0.483
Multiclass AraBERTv0.2-Twitter	0.490
Multiclass Ensemble	0.510
Hierarchical MARBERTv2	0.538
Hierarchical AraBERTv0.2-Twitter	0.547
Hierarchical Ensemble	0.553

Table 4: Performance comparison for Subtask 2 across configurations on test set.

6 Conclusion

In this paper, we presented our system for the MAHED 2025 Shared Task, which leverages Arabic-specific BERT ensembles with soft voting and a hierarchical cascaded pipeline for Subtask 2 to detect hope and hate emotions in Arabic content. Our approach achieved competitive results, demonstrating the effectiveness of ensemble methods and emotion augmentation in handling class imbalances and hierarchical dependencies.

Several promising directions emerge for enhancing system performance: implementing targeted data augmentation strategies for underrepresented classes, incorporating large language models (LLMs) to leverage their contextual understanding capabilities to more effectively address class imbalances.

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.

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmad, Md. Arid Hasan, and Maram Hasanain. 2024b. ArMeme: Propagandistic content in Arabic memes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21071–21090, Miami, Florida, USA. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Md. Rafiul Biswas and Wajdi Zaghouani. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *CoRR*, abs/2505.11969.

Md. Rafiul Biswas and Wajdi Zaghouani. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *CoRR*, abs/2505.11959.

Maram Hasanain, Fatema Ahmad, and Firoj Alam. 2024a. Large language models for propaganda span annotation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14522–14532, Miami, Florida, USA. Association for Computational Linguistics.

Maram Hasanain, Fatema Ahmed, and Firoj Alam. 2024b. Can gpt-4 identify propaganda? annotation and detection of propaganda spans in news articles. In *Proceedings of the 2024 Joint International Conference On Computational Linguistics, Language Resources And Evaluation*, LREC-COLING 2024, Torino, Italy.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024a. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024b. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

ANLPers at MAHED Shared Task: From Hate to Hope: Boosting Arabic Text Classification

Yasser Alhabashi[1] **Serry Sibaee**[1*] **Omer Nacar**[2] **Adel Ammar**[1] **Wadii Boulila**[1]
[1]Prince Sultan University, Riyadh, Saudi Arabia
[2]Tuwaiq Academy – Tuwaiq Research and Development Center
{yalhabashi, ssibaee, aammar , wboulila}@psu.edu.sa
{o.najar}@tuwaiq.edu.sa
*Corresponding author: ssibaee@psu.edu.sa

Abstract

The detection of harmful online content, including hate speech and propaganda, is particularly challenging in multimodal and multilingual contexts such as Arabic social media. This work addresses **Sub-task 1: Text-based Hate and Hope Speech Classification** in the MAHED2025 (Zaghouani et al., 2025) challenge, where the goal is to classify Arabic text into *hate*, *hope*, or *not_applicable*. We develop a system based on pre-trained Arabic BERT models with three fine-tuning strategies, combined with a custom preprocessing pipeline for noise removal, normalization, and diacritic stripping. To address class imbalance and lexical sparsity, we augment the training data with synthetically generated paraphrases via the OpenAI API. Experimental results on the official test set demonstrate that our best configuration, **BERT-base-AraBERTv02 + NN** with cleaning and generated data, achieves a macro-F1 score of **0.6747** F1. Error analysis reveals that mislabeled training instances significantly limit model performance, suggesting that future improvements may be achieved through systematic dataset refinement. Our approach highlights the importance of preprocessing, augmentation, and careful architectural choices for robust Arabic text classification.

1 Introduction

The rapid growth of social media has transformed the way information is produced, shared, and consumed, enabling unprecedented reach and immediacy. However, this openness has also facilitated the large-scale dissemination of harmful content such as hate speech, propaganda, and other forms of toxic communication. While such material may appear in text, images, or videos, multimodal formats like memes present a unique challenge for automated detection due to their combination of linguistic and visual cues, cultural references, and implicit meanings (Alam et al., 2024). These challenges are further compounded when hateful or propagandistic elements are intertwined, requiring models to capture subtle contextual overlaps between intent, emotion, and target.

Existing research has made significant progress in detecting harmful content across various modalities, languages, and levels of granularity. For instance, several studies have focused on annotating and analyzing large datasets for hate speech, offensive language, and related emotional attributes, particularly in underrepresented languages such as Arabic [(Zaghouani et al., 2024a),(Zaghouani and Biswas, 2025b)]. Others have highlighted the need to move beyond binary classification toward multi-label frameworks that capture target type, severity, and overlapping categories (Alam et al., 2024). Despite these advances, a number of persistent issues hinder progress: small and heterogeneous datasets, low inter-annotator agreement, inconsistent evaluation methodologies, and model performance drops when applied across domains or languages (Bäumler et al., 2025).

Moreover, most prior work treats modalities in isolation—either text-only or image-only—leaving limited exploration of their intersection, especially in contexts where textual and visual signals work jointly to convey harmful messages (Zaghouani et al., 2024a). Multilingual and cross-linguistic challenges remain especially acute, with the scarcity of high-quality annotated datasets further complicating model development (Zaghouani and Biswas, 2025b). Additionally, while transformer-based models such as BERT have shown strong performance in single-modality tasks (Bäumler et al., 2025), their application in complex, multimodal, multi-label scenarios remains underexplored.

Our work addresses these challenges through a novel approach that integrates advanced NLP pre-processing techniques with BERT-based model training, enabling more accurate and nuanced detection of harmful multimodal content. By leveraging fine-tuned linguistic preprocessing to normalize

Proceedings of The Third Arabic Natural Language Processing Conference, pages 590–594
November 8-9, 2025 ©2025 Association for Computational Linguistics

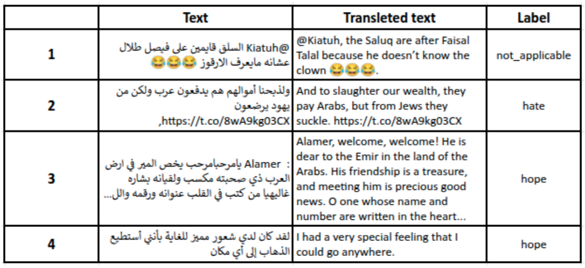
and enrich textual data before BERT training, we improve the model's ability to capture subtle semantic and contextual cues that are often missed in raw text. This combination not only enhances classification accuracy in multi-label settings but also facilitates better generalization across different domains and linguistic varieties. In doing so, our approach bridges critical gaps identified in the literature and provides a scalable pathway toward more robust and context-aware harmful content detection systems.

2 Background

We use the Arabic hate/hope speech dataset introduced by (Zaghouani et al., 2024b; Zaghouani and Biswas, 2025c,a) as part of the **Sub-task 1: Text-based Hate and Hope Speech Classification** in the MAHED2025 shared task (Zaghouani et al., 2025). The goal is to classify Arabic text—either Modern Standard Arabic (MSA) or dialectal—into one of three categories:

- **Hate**: Hostile, offensive, or discriminatory content.
- **Hope**: Optimistic, encouraging, or positive sentiment.
- **Not Applicable**: Neutral text without hate or hope signals.

The input is an Arabic sentence, and the output is a label from the set {hate, hope, not_applicable}. Below (in Table 1) are example instances from the dataset, with their corresponding labels:



	Text	Transleted text	Label
1	@Kiatuh السلق قايمين على فيصل طلال عشانه مايعرف الارقوز 😂😂😂	@Kiatuh, the Saluq are after Faisal Talal because he doesn't know the clown 😂😂😂.	not_applicable
2	ولذبحنا أموالهم هم يدفعون عرب ولكن من يهود يرضعون ,https://t.co/8wA9kg03CX	And to slaughter our wealth, they pay Arabs, but from Jews they suckle. https://t.co/8wA9kg03CX	hate
3	: Alamer يامرحبامرحب يخص المير في ارض العرب ذي صحبته مكسب ولقيانه بشاره غاليهيا من كتب في القلب عنوانه ورقمه والل...	Alamer, welcome, welcome! He is dear to the Emir in the land of the Arabs. His friendship is a treasure, and meeting him is precious good news. O one whose name and number are written in the heart...	hope
4	لقد كان لدي شعور مميز للغاية بأنني أستطيع الذهاب إلى أي مكان	I had a very special feeling that I could go anywhere.	hope

Table 1: Dataset instances.

The original training set contains **6,890** instances with columns `text` and `label`. A class distribution analysis reveals a moderate imbalance toward the *not_applicable* class, with a majority/minority ratio of approximately **2.84**. Table 2 shows the distribution.

Label	Count	Percent
hate	1,301	18.88%
hope	1,892	27.46%
not_applicable	3,697	53.66%
Total	6,890	100%

Table 2: Class distribution in the original dataset.

Texts in the dataset average **22.48** words (median **18**; 95th percentile **54**) and **139.64** characters (median **109.5**; 95th percentile **357**). These figures indicate that most inputs are relatively short, but there is a long tail of longer utterances. The observed imbalance motivates the use of macro-averaged metrics for evaluation and, during training, class-aware strategies such as re-weighting or targeted augmentation to improve model robustness across all categories.

3 System Overview

Our system is built upon pre-trained transformer-based Arabic language models, with multiple fine-tuning strategies. We explored three main architectures:

Variant A: BERT as Frozen Embeddings + Neural Network. We freeze the BERT encoder (Sibaee et al., 2024), compute average-pooled sentence embeddings, and train a feed-forward neural network. Two configurations were tested: one with 8 layers. All hidden layers use GELU activations and optional batch normalization.

Variant B: Fine-tuning BERT End-to-End. We fine-tune the BERT model directly for the classification task by updating all encoder parameters during training. A single linear classification head is applied on top of the pooled sentence representation.

Variant C: Fine-tuning BERT + Additional Fully Connected Layers. We fine-tune the BERT encoder and append two additional fully connected layers before the classification layer. These layers use GELU activations and optional batch normalization to capture higher-level abstractions.

All models incorporate our cleaning pipeline, which removes Latin characters, symbols, emojis, and Arabic diacritics, normalizes Unicode, and collapses extra spaces.

4 Experimental Setup

4.1 Data

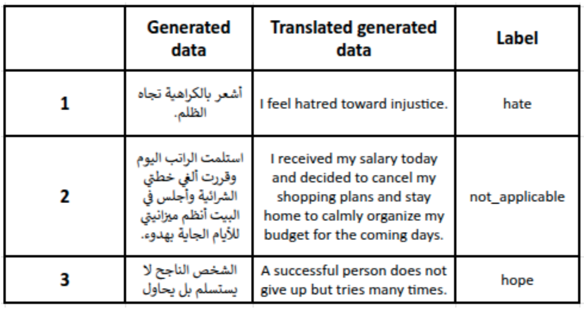
We evaluate our models under three data preparation settings:

1. **Without Cleaning:** raw text as provided in the original dataset.
2. **With Cleaning:** applying the custom preprocessing function described in Section 4.2.
3. **With Cleaning + Generated Data:** combining cleaned text with synthetically generated paraphrases to increase lexical diversity.

Without augmentation, the training set contains 1,000 samples per class (3,000 total) and the validation set contains 250 samples per class (750 total). With generated data, the training set grows to 4,000 samples per class (12,000 total) and the validation set includes 300 samples per class (900 total). The official test set is provided by the task organizers.

Synthetic Data Generation. To address class imbalance and enhance linguistic diversity, we expanded the training set with **synthetically generated paraphrases** of existing samples. Paraphrases were produced using the **GPT4-mini** (OpenAI et al., 2024), guided by prompts designed to generate semantically equivalent Arabic sentences while preserving the original class labels. The generation process introduced lexical, structural, and stylistic variations without altering the underlying meaning, enabling the model to better generalize to unseen expressions.

table 3 presents examples of generated sentences alongside their corresponding labels.



	Generated data	Translated generated data	Label
1	أشعر بالكراهية تجاه الظلم.	I feel hatred toward injustice.	hate
2	استلمت الراتب اليوم وقررت ألغي خطتي الشرائية وأجلس في البيت أنظم ميزانيتي للأيام الجاية بهدوء.	I received my salary today and decided to cancel my shopping plans and stay home to calmly organize my budget for the coming days.	not_applicable
3	الشخص الناجح لا يستسلم بل يحاول	A successful person does not give up but tries many times.	hope

Table 3: Examples of synthetically generated Arabic data with corresponding labels.

4.2 Preprocessing

The custom text cleaning pipeline performs the following steps:

- Remove non-Arabic letters.
- Remove punctuation symbols.
- Remove emojis and pictographs.
- Remove Arabic diacritics.
- Remove diacritics from other languages via Unicode normalization.

Finally, multiple spaces are collapsed into a single space, preserving the core Arabic words.

4.3 Training Details

We use the AdamW optimizer with a linear decay learning rate schedule and warmup. Learning rates tested across experiments include 1×10^{-4}, 2×10^{-5}, 1×10^{-5}, and 1×10^{-6}. The batch size is fixed at 32. Early stopping is applied with a patience of 10 to prevent overfitting; no fixed epoch count is used. For most experiments, we use a dropout rate of 0.3, while for **Variant C** we additionally test a higher dropout rate of 0.7.

5 Results

Results are reported using the official evaluation metric (average macro-F1-score). Table 4 presents the validation and test **average macro-F1-score** for all model variants under the three data preparation settings. Our best **test** result is **0.6747**, achieved with **BERT-base-AraBERTv02 (Antoun et al., 2021) + NN** using cleaning, generated data, and a learning rate of 2×10^{-5}.

6 Error Analysis

Despite achieving competitive macro-F1 scores, our models' performance is limited by annotation quality. A manual review of a subset of the training data revealed a substantial proportion of mislabeled instances, which can mislead the learning process and reduce model generalization.

6.1 Quantitative Error Breakdown

We manually evaluated a random sample of 100 training examples. Out of these, 78 samples were correctly labeled, while 22 (22%) were found to be mislabeled. The dataset is heavily skewed toward the *not_applicable* class, followed by *hope* and *hate*, as shown in Figure 1. Figure 2 illustrates the number of correct vs. mislabeled samples for each class.

Model	LR	Clean	Gen. Data	Dropout	Val Avg. Macro-F1	Test Avg. Macro-F1
BERT-base-AraBERTv02	1e-5	Yes	Yes	0.3	0.6281	0.6504
BERT-base-AraBERTv02 + NN	2e-5	Yes	No	0.7	0.6386	0.6736
BERT-base-AraBERTv02 + NN	2e-5	Yes	Yes	0.3	0.6394	**0.6747**
BERT-base-AraBERTv02 embd + 8 layers	2e-5	Yes	Yes	0.3	0.5863	0.6235

Table 4: Comparison of experimental settings and corresponding validation/test macro-F1 scores. Settings are shown first for clearer interpretability.

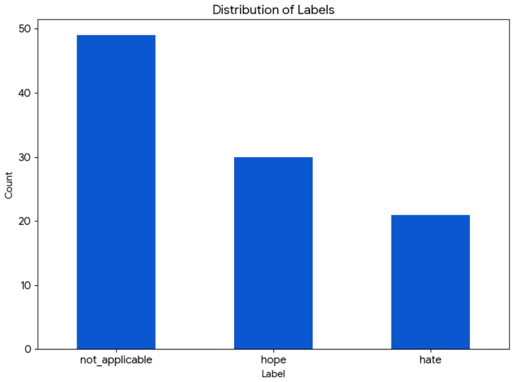


Figure 1: Distribution of studied samples

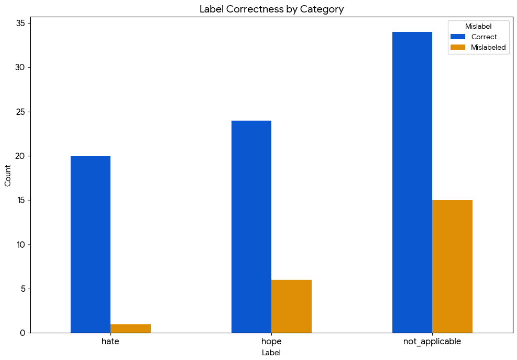


Figure 2: Label correctness by category, showing the number of correctly labeled vs. mislabeled samples per class.

6.2 Label Quality Summary

Figure 2 summarizes the distribution of correctly labeled vs. mislabeled samples by true class. While all three categories are affected by labeling errors, 'not_applicable' exhibits the highest mislabel rate relative to its class size (8 of 29 samples, $\sim$27.6%). Nearly a quarter of the reviewed data was mislabeled, highlighting that annotation noise is a major bottleneck. These findings suggest that systematic dataset relabeling or consensus-based annotation is crucial to improving model robustness (Sibaee et al., 2025), showing in Table 5 after correcting the labels for each category.

Labels	Correctly Labeled	After Correction	Total
Hate	21	8	29
Hope	30	7	37
Not_applicable	49	7	56
Total	78	22	100

Table 5: Breakdown of correctly labeled and mislabeled samples per true class in the manually reviewed subset, and after correcting each category.

7 Conclusion

In this paper, we introduced a BERT-based Arabic text classification system developed for the MAHED2025: Task-1 challenge, integrating tailored preprocessing, synthetic data generation, and multiple fine-tuning strategies. Our best configuration, combining AraBERT embeddings with additional neural network layers and generated data, achieved a macro-F1 score of 0.6747, demonstrating the effectiveness of our approach. However, manual error analysis revealed a considerable proportion of mislabeled instances in the dataset, which limits performance even with advanced models. Future work will focus on improving annotation quality through re-labeling or consensus-based methods, as well as exploring domain adaptation, cross-lingual transfer, and multimodal extensions to build more accurate, robust, and context-aware systems for harmful content detection in underrepresented languages like Arabic.

Acknowledgments

We would like to thank Prince Sultan University for their generous support in enabling this research.

References

Firoj Alam, Md. Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024. Propaganda tonbsp;hate: A multimodal analysis ofnbsp;arabic memes withnbsp;multi-agent llms. In *Web Information Systems Engineering – WISE 2024: 25th International Conference, Doha, Qatar, December 2–5,* *2024, Proceedings, Part V*, page 380–390, Berlin, Heidelberg. Springer-Verlag.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Arabert: Transformer-based model for arabic language understanding. *Preprint*, arXiv:2003.00104.

Julian Bäumler, Louis Blöcher, Lars-Joel Frey, Xian Chen, Markus Bayer, and Christian Reuter. 2025. A survey of machine learning models and datasets for the multi-label classification of textual hate speech in english. *Preprint*, arXiv:2504.08609.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Serry Sibaee, Abdullah Alharbi, Samar Ahmad, Omer Nacar, Anis Koubaa, and Lahouari Ghouti. 2024. ASOS at KSAA-CAD 2024: One embedding is all you need for your dictionary. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 697–703, Bangkok, Thailand. Association for Computational Linguistics.

Serry Sibaee, Omer Nacar, Adel Ammar, Yasser Al-Habashi, Abdulrahman Al-Batati, and Wadii Boulila. 2025. From guidelines to practice: A new paradigm for arabic language model evaluation. *Preprint*, arXiv:2506.01920.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md. Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *Preprint*, arXiv:2505.11959.

Wajdi Zaghouani and Md Rafiul Biswas. 2025c. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024a. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024b. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

LoveHeaven at MAHED 2025: Text-based Hate and Hope Speech Classification Using AraBERT-Twitter Ensemble

Nguyen Thien Bao, Dang Van Thin
University of Information Technology-VNUHCM,
Vietnam National University, Ho Chi Minh City, Vietnam
23520127@gm.uit.edu.vn
thindv@uit.edu.vn

Abstract

This paper presents our system for Sub-task 1 in MAHED 2025 (Zaghouani et al., 2025) shared task: Text-based Hate and Hope Speech Classification. We propose a robust pipeline built upon the bert-base-arabertv02-twitter model, leveraging domain-specific preprocessing, hyperparameter optimization with Optuna, and a K-Fold ensemble strategy. This system ranked 4 th among all participating teams on the leaderboard. We discuss technical design choices, the results of ablation studies, and the impact of preprocessing and model selection on final performance.

1 Introduction

Social media in the Arabic-speaking world exhibits a dynamic interplay between hateful and hopeful expressions, often entangled with rich dialectal diversity, code-switching, and informal orthography that complicate automatic detection. Beyond text, hateful content is increasingly conveyed via multimodal artifacts such as memes (Alam et al., 2024), motivating systems capable of analyzing both textual and visual modalities. Within this context, MAHED 2025 (Zaghouani et al., 2025) is organized as a shared task at ArabicNLP 2025 (co-located with EMNLP 2025), covering hope/hate and emotion detection in single-task, multi-task, and multimodal settings.

This paper presents a text-only system for Sub-task 1, where the input is Arabic text (MSA or dialect) and the output is one of three labels: *hate*, *hope*, or *not_applicable*. The task evaluates systems by macro-averaged F1, a metric robust under class imbalance.

2 Related Work

Pre-trained transformer models for Arabic, notably AraBERT (Antoun et al., 2020), have established strong baselines on sentiment, dialect identification, and harmful content detection. AraBERTv0.2-Twitter (Antoun et al., 2020) extends this by further pretraining on a large corpus of tweets to better handle dialectal and informal Arabic. Recent datasets for harmful, offensive, and hopeful Arabic speech (Zaghouani and Biswas, 2025a; Zaghouani et al., 2024; Zaghouani and Biswas, 2025b) highlight the need for balanced evaluation metrics like macro-F1. For multimodal hateful content, studies such as (Alam et al., 2024) show the value of multimodal fusion techniques.

3 Background

3.1 Task Setup

Sub-task 1 requires a three-way classification: *hate*, *hope*, and *not_applicable*, for short Arabic text. The evaluation uses macro-F1 to handle class imbalance. In particular, the validation and test labels are concealed from participants. Predictions must be submitted to the official leaderboard to obtain macro-F1 scores, promoting strong generalization and preventing tuning on these datasets.

3.2 SubTask1 and its dataset

Sub-task 1 is a three-way classification problem: *hate*, *hope*, *not_applicable*. Input is short Arabic text in MSA or dialect. The dataset (Zaghouani et al., 2024) includes contributions from multiple platforms, with annotations performed manually by native speakers. Training set: 6,890 labeled samples; validation set: 1,476 unlabeled. Evaluation uses macro-F1 as the primary metric.

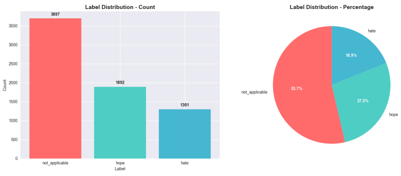


Figure 1: Label Distribution in training data

Key dataset observations:

- Quite imbalanced label distribution but can be acceptable (Figure 1).
- Short, noisy social-media texts — suitable for 128–256 BERT token length. After evaluating both configurations on the validation and test data, we found that a token length of 256 is more suitable for our pipeline in this task, providing better performance and results.(Figure 2)

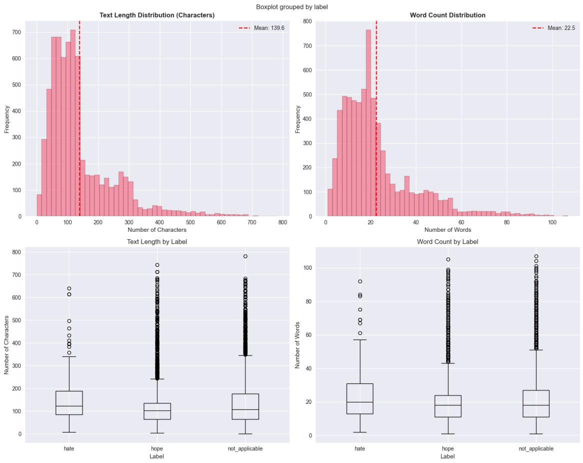


Figure 2: Text length distribution

4 System Overview

4.1 Duplicate handling

The dataset is of high quality with no missing values but has around 320 duplicate entries found in the training set.
Duplicate handling in the training dataset:

- Same text, conflicting labels → remove all.
- Same text, same label → keep one text.

The final label distribution after handling duplicate (Figure 3)

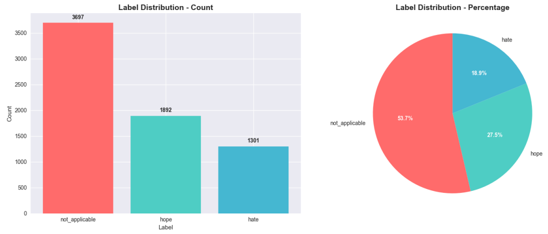


Figure 3: Label distribution after handling duplicate training set

4.2 Preprocessing

We import ArabertPreprocessor from arabert.preprocess for automatically handles (Antoun et al., 2020):

- Text normalization.
- Remove non-Arabic characters, URLs, mentions.
- Tokenize via HuggingFace AutoTokenizer (max_len=256).

4.3 Model and System

Our approach uses aubmindlab/bert-base-arabertv02-twitter (Antoun et al., 2020), pretrained on ∼60M tweets, alongside Arabic-aware preprocessing and Optuna-driven hyperparameter tuning. A 4-fold ensemble is used for robustness.(Figure 4)

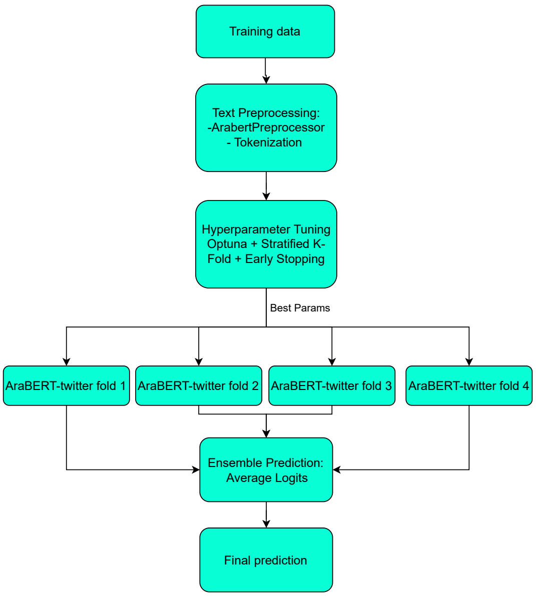

Training data
Text Preprocessing: -ArabertPreprocessor - Tokenization
Hyperparameter Tuning Optuna + Stratified K-Fold + Early Stopping
Best Params
AraBERT-twitter fold 1
AraBERT-twitter fold 2
AraBERT-twitter fold 3
AraBERT-twitter fold 4
Ensemble Prediction: Average Logits
Final prediction

Figure 4: Pipeline of Technique in subtask 1

After having Best parameters from Optuna, the dataset was split into 4 folds using StratifiedKFold. We trained 4 separate AraBERT models from scratch, one for each fold, using the best parameters found by Optuna. The inference is based on the average logits from all folds.

5 Experimental Setup

5.1 Resources

We trained and evaluated all models on Kaggle Notebooks (free tier) with a single NVIDIA Tesla

P100 16GB GPU, 2 vCPUs, and approximately 13GB RAM. The environment used PyTorch 2.6.0, Transformers 4.52.4, and Optuna 4.4.0 on the default Kaggle Linux image (Python 3.11). Training sessions were constrained by free-tier time limits; each fold completed within a single session.

5.2 Hyperparameter Search

The Optuna experiment was run with 18 trials with tokenizer max_length = 256 and 30 trials with max_length = 128, both using early stopping (patience = 3). Due to resource constraints, the experiment completed all trials before the early stopping criteria were met. After evaluating on the validation and test data, we chose max_length = 256 with Optuna (n_trials = 18) as the better choice for our pipeline.
Loss: cross-entropy.
Optimizer: AdamW.
Scheduler: linear warmup-decay.
Hyperparameter tuning via Optuna:

- Learning Rate $\in [1 \times 10^{-6}, 1 \times 10^{-5}]$
- Batch size $\in \{8, 16, 32\}$
- Epochs $\in [2, 5]$

Best parameters: Learning Rate $\approx 9.74 \times 10^{-6}$, batch=8, epochs=4.

6 Results

We experimented with two state-of-the-art Arabic BERT models from the aubmindlab repository using the same pipeline.

Metric	AraBERT-Twitter	AraBERT
F1	0.6563	0.6403
Accuracy	0.6775	0.6511
Precision	0.6600	0.6343
Recall	0.6533	0.6477

Table 1: Comparison of AraBERT and AraBERT-Twitter on Validation data

Due to its higher F1-score on the validation data (0.66 compared to 0.64), the arabertv02-twitter model was selected for the final pipeline. Its specialization in social media text is particularly relevant to the dialectal and informal nature of the dataset. Moreover, the arabertv02-twitter model also outperformed the other model on the test data.

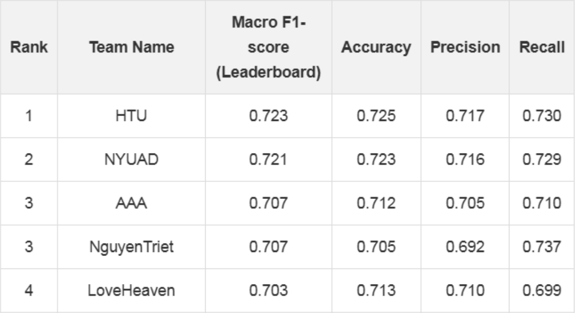
Metric	AraBERT-Twitter	AraBERT
F1	0.7030	0.7017
Accuracy	0.7130	0.7109
Precision	0.7100	0.7061
Recall	0.6990	0.6982

Table 2: Comparison of AraBERT and AraBERT-Twitter on Test data

To evaluate the impact of the ensemble approach, we compared our 4-fold StratifiedKFold ensemble against training a single `aubmindlab/bert-base-arabertv02-twitter` model on the full training set using the same best hyperparameters found via Optuna. The single-model setup slightly underperforms in the validation and test dataset compared to the ensemble, suggesting that ensembling mitigates variance and improves robustness, particularly under class imbalance conditions. This aligns with our observation that different folds capture complementary patterns in the training data.

6.1 Leaderboard

Our system ranked 4th on the official competition Leaderboard, with a Macro F1 score just 0.02 behind the top-ranked team. Our Accuracy and Precision placed us in the top 3, while our competitive recall (0.699) secured a position in the top 4. This result showcases a quite strong overall performance.



Rank	Team Name	Macro F1-score (Leaderboard)	Accuracy	Precision	Recall
1	HTU	0.723	0.725	0.717	0.730
2	NYUAD	0.721	0.723	0.716	0.729
3	AAA	0.707	0.712	0.705	0.710
3	NguyenTriet	0.707	0.705	0.692	0.737
4	LoveHeaven	0.703	0.713	0.710	0.699

Figure 5: The Final Leaderboard by macro-F1

7 Conclusion

We presented a competitive text-only system for MAHED 2025 Sub-task 1, ranking 4th by macro-F1 on the Leaderboard. In conclusion, our proposed AraBERT-based ensemble framework, optimized with Stratified K-Fold and Optuna for macro-F1, demonstrates significant effectiveness in classifying Arabic text into hate, hope, and not_applicable categories, highlighting the potential of transformer-based models combined with ensemble learning for nuanced emotion detection in low-resource languages.

7.1 Limitations

This work uses text-only inputs; multimodal cues from images/memes are not modeled. Dialectal diversity and code-switching can reduce recall on minority or subtle cases, especially *hope* vs *not_applicable*. Label subjectivity around borderline cases can introduce noise across folds. Resource constraints (free-tier Kaggle Notebooks) limited the breadth of hyperparameter exploration.

7.2 Ethical Considerations

Misclassifying harmful content as benign can cause user harm and under-enforcement; human-in-the-loop moderation is recommended in high-stakes deployments. Data derived from social media may contain sensitive content and PII; usage should respect licensing, privacy, and minimize potential disparate impacts on dialect communities.

7.3 Future work

Future work includes expanding to multimodal inputs (images/memes), stronger dialect handling, and uncertainty-aware inference.

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

A Appendix

Full hyperparameters and code are available at: https://github.com/Limdim1604/MAHED2025

CIC-NLP at MAHED 2025 TASK 1:Assessing the Role of Bigram Augmentation in Multiclass Arabic Hate and Hope Speech Classification

Obiadoh A. E., Abiola O.J, Ogunleye T.D., Tewodros B.A.[1], Abiola T.O[1],
[1]Instituto Politecnico Nacional, Centro de Investigacion en Computacion, CDMX, Mexico.,
Correspondence: tabiola2025@cic.ipn.mx

Abstract

This study investigates the impact of bigram-based data augmentation on the joint classification of hate speech, hope speech, and neutral content in multilingual social media contexts, with a particular focus on Arabic. While previous research has shown the benefits of augmentation in text classification, its effectiveness in nuanced domains such as hate and hope speech remains underexplored. Using the annotated MAHED dataset, we compare three scenarios: a baseline without augmentation, global bigram augmentation, and classwise bigram augmentation. The baseline achieved 68.25% accuracy (macro-F1 = 0.6729) on the test set. Global bigram augmentation slightly reduced accuracy to 63.0% (macro-F1 = 0.62), showing no improvement over the baseline. Classwise augmentation achieved 93% accuracy on the validation set but dropped sharply to 59.65% accuracy (macro-F1 = 0.4726) on the test set, indicating severe overfitting. These results suggest that bigram-based methods are sensitive to class imbalance and may harm generalisation when applied unevenly across classes. We conclude by highlighting the need for more balanced, context-aware augmentation strategies in socially impactful NLP tasks.

1 Introduction

Hate speech and hope speech represent two critical yet contrasting forms of online expression. Hate speech fosters hostility, discrimination, and division (Alshahrani et al., 2025; **?**), while hope speech promotes unity, resilience, and positive social change (**?**). With the rapid growth of social media platforms, especially in multilingual and dialect-rich contexts such as Arabic, the automatic detection of these speech forms has become a pressing challenge. Although hate speech detection has received significant research attention (Al-Sukhani et al., 2025; Gasmi et al., 2025), hope speech detection remains comparatively underexplored, and the combined classification of both introduces unique complexities. These challenges include linguistic diversity, scarcity of high-quality annotated datasets, and the nuanced cultural and contextual variations in language use (Alrasheed et al., 2025).

Data augmentation has emerged as a promising strategy to improve the robustness and generalisation of natural language processing models, particularly in low-resource scenarios. Among these, bigram-based augmentation methods have shown success in enhancing text classification performance by enriching contextual co-occurrence patterns. However, their efficacy in nuanced, multi-class problems—such as joint hate and hope speech classification—remains uncertain. In this study, we investigate the impact of different bigram augmentation strategies, namely global and classwise augmentation, in comparison with a non-augmented baseline. Through a comprehensive empirical evaluation, we identify scenarios where augmentation may fail to deliver expected gains and discuss the implications for future work in socially impactful NLP applications.

2 Background

Recent advances in text classification have been driven by the adoption of Large Language Models (LLMs) across diverse domains. Early transformer-based approaches showed strong performance on complex linguistic tasks (Kolesnikova and Gelbukh, 2020; Adebanji et al., 2022), while more recent studies have explored fine-tuning and prompt-based methods for low-resource and multilingual contexts (Abiola et al., 2025c,b). Shared tasks and benchmarks (Ojo et al., 2023; Achamaleh et al., 2025) have further tested LLM robustness in noisy, real-world settings, and other works (Oladepo et al., 2025; Abiola et al., 2025a) have integrated contextual cues to improve predictive performance.

In the context of Arabic hate and hope speech

Proceedings of The Third Arabic Natural Language Processing Conference, pages 599–602
November 8-9, 2025 ©2025 Association for Computational Linguistics

detection, challenges arise from dialectal diversity, morphological richness, and scarcity of annotated resources. The MAHED shared task (Zaghouani et al., 2025) addresses this by providing a labelled dataset with three categories: *hate*, *hope*, and *not_applicable*, encouraging participants to explore robust, generalisable classification approaches. Our submission focuses on a MARBERT-based pipeline with hybrid lexical–contextual augmentation via bigrams.

3 System Overview

Our system combines light preprocessing, a transformer encoder (MARBERT), and three bigram augmentation strategies. We use MARBERT (`UBC-NLP/MARBERT`) to capture deep contextual semantics and append frequent bigrams as explicit lexical cues. This design addresses two key challenges: (1) dialectal variation, by using MARBERT's pretraining coverage, and (2) sparse surface features, by injecting high-frequency n-grams into the input.

3.1 Preprocessing

We normalise Arabic text with the `ArabertPreprocessor` (AraElectra profile), preserving emojis to retain affective cues. No morphological segmentation is applied.

3.2 Bigram Augmentation

We explore:

- **Global-top**: top-K bigrams across the corpus, appended to all samples.
- **Class-specific**: top-K bigrams per class, appended based on ground-truth labels.
- **Unsupervised test-time**: predicted dominant class bigrams appended using overlap heuristics.

3.3 Training Setup

We compare:

1. **Baseline**: MARBERT with no augmentation (10 epochs).
2. **Hybrid**: MARBERT with bigram-augmented text (4 epochs).

Training uses AdamW (HuggingFace defaults), batch size = 16, maximum sequence length = 128, and model selection by validation macro-F1.

Class	Precision	Recall	F1	Support
0	0.59	0.63	0.61	238
1	0.62	0.55	0.58	359
2	0.69	0.71	0.70	729

Table 1: Validation metrics — Baseline.

Class	Precision	Recall	F1	Support
0	0.53	0.69	0.60	238
1	0.62	0.57	0.59	359
2	0.69	0.65	0.67	729

Table 2: Validation metrics — Global bigram augmentation.

4 Experimental Setup

The MAHED dataset is split into `train`, `val`, and `test` as per organisers. Labels are encoded via `LabelEncoder` for consistency. Evaluation metric: macro-F1 (primary), along with accuracy, precision, and recall.

5 Results

5.1 Validation Performance

The baseline achieved macro-F1 $= 0.63$ (accuracy $= 0.65$), with the majority class performing best. Global bigrams improved minority-class recall but reduced majority-class accuracy. Classwise bigrams yielded extremely high validation performance (macro-F1 $= 0.92$) but failed to generalise.

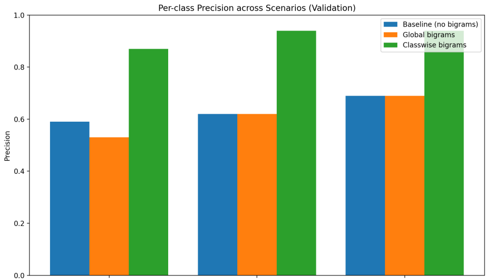


Figure 1: Per-class precision (validation).

5.2 Test Performance and Generalisation

The baseline maintained macro-F1 $= 0.6729$ on test data, while classwise bigrams dropped sharply to 0.4726 due to overfitting.

5.3 Error Analysis

Global bigrams: Provided minor recall gains for minority classes but reduced precision for the majority class.

Class	Precision	Recall	F1	Support
0	0.87	0.87	0.87	238
1	0.94	0.95	0.95	359
2	0.94	0.94	0.94	729

Table 3: Validation metrics — Classwise bigram augmentation.

Scenario	Accuracy	Precision	Recall	Macro-F1
Baseline (test)	0.6825	0.6742	0.6733	0.6729
Classwise bigrams (test)	0.5965	0.6802	0.4660	0.4726

Table 4: Test metrics: Baseline vs. Classwise bigrams.

Classwise bigrams: Boosted validation scores artificially by memorising label-specific tokens, which became noise in test scenarios.

Other factors: Token truncation and domain shift likely reduced augmentation benefits.

6 Conclusion

Global bigram augmentation offered only small gains, while classwise augmentation inflated validation results but failed in generalisation. This underscores the risk of label-tied augmentation in imbalanced, nuanced datasets and points to the need for label-agnostic, domain-robust augmentation strategies.

Acknowledgments

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20241816, 20241819, and 20240951 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

7 Limitations

The small, imbalanced dataset may have skewed augmentation effects, with classwise augmentation risking overfitting for rare classes. We only tested bigram-based methods, leaving other strategies (e.g., paraphrasing, back-translation, contextual augmentation) unexplored. Evaluation was confined to in-domain data, so cross-domain and cross-dialect generalisation is uncertain. Finally, we did not assess interpretability, which is important to prevent augmentation-induced bias.

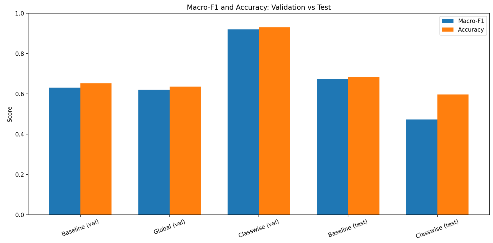


Figure 2: Macro-F1 and accuracy for validation and test.

Acknowledgments

References

Tolulope Abiola, Olumide Ebenezer Ojo, Grigori Sidorov, Olga Kolesnikova, and Hiram Calvo. 2025a. CIC-IPN at SemEval-2025 task 11: Transformer-based approach to multi-class emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1609–1615, Vienna, Austria. Association for Computational Linguistics.

Tolulope O. Abiola, Tewodros A. Bizuneh, Oluwatobi J. Abiola, Temitope O. Oladepo, Olumide E. Ojo, Adebanji O. O., Grigori Sidorov, and Olga Kolesnikova. 2025b. Cic-nlp at genai detection task 1: Leveraging distilbert for detecting machine-generated text in english. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, Abu Dhabi, UAE. Association for Computational Linguistics.

Tolulope O. Abiola, Tewodros A. Bizuneh, Fatima Uroosa, Nida Hafeez, Grigori Sidorov, Olga Kolesnikova, and Olumide E. Ojo. 2025c. Cic-nlp at genai detection task 1: Advancing multilingual machine-generated text detection. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, Abu Dhabi, UAE. Association for Computational Linguistics.

Tewodros Achamaleh, Tolulope Olalekan Abiola, Lemlem Eyob Kawo, Mikiyas Mebraihtu, and Grigori Sidorov. 2025. CIC-NLP@DravidianLangTech 2025: Detecting AI-generated product reviews in Dravidian languages. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 502–507, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Olaronke Oluwayemisi Adebanji, Irina Gelbukh, Hiram Calvo, and Olumide Ebenezer Ojo. 2022. Sequential models for sentiment analysis: A comparative study. In *Advances in Computational Intelligence, 21st Mexican International Conference on Artificial*

Intelligence, MICAI 2022, Proceedings, Part II, Monterrey, Mexico. Springer.

Hassan Al-Sukhani, Qusay Bsoul, Abdelrahman H. Elhawary, Ziad M. Nasr, Ahmed E. Mansour, Radwan M. Batyha, Basma S. Alqadi, Jehad Saad Alqurni, Hayat Alfagham, and Magda M. Madbouly. 2025. Multilingual hate speech detection: Innovations in optimized deep learning for english and arabic hate speech detection. *SN Computer Science*, 6(205).

Sadeem Alrasheed, Suliman Aladhadh, and Abdulatif Alabdulatif. 2025. Protecting intellectual security through hate speech detection using an artificial intelligence approach. *Algorithms*, 18(4):179.

Eman S. Alshahrani, Mehmet S. Aksoy, and Ahmed Emam. 2025. Detection of hate speech and offensive language in arabic text: A systematic literature review. *Applied Computational Intelligence and Soft Computing*. First published: 13 April 2025.

Karim Gasmi, Ibtihel Ben Ltaifa, Alameen Eltoum Abdalrahman, Omer Hamid, Mohamed Othman Altaieb, and Shahzad Ali. 2025. Hybrid feature and optimized deep learning model fusion for detecting hateful arabic content. *IEEE Access*, 13.

O. Kolesnikova and A. Gelbukh. 2020. A study of lexical function detection with word2vec and supervised machine learning. *J. Intell. Fuzzy Syst.*, 39.

Olumide E. Ojo, Olaronke O. Adebanji, Hiram Calvo, Damian O. Dieke, Olumuyiwa E. Ojo, Seye E. Akinsanya, Tolulope O. Abiola, and Anna Feldman. 2023. Legend at araieval shared task: Persuasion technique detection using a language-agnostic text representation model. *Preprint*, arXiv:2310.09661.

Temitope Oladepo, Oluwatobi Abiola, Tolulope Abiola, Abdullah , Usman Muhammad, and Babatunde Abiola. 2025. Predicting emotion intensity in text using transformer-based models. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1677–1682, Vienna, Austria. Association for Computational Linguistics.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgio Mikros, Abul Hasnat, and Firoj Alam. 2025. Overview of mahed shared task: Multimodal detection of hope and hate emotions in arabic content. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

A Example Appendix

This is an appendix.

TranTranUIT at MAHED Shared Task: Multilingual Transformer Ensemble with Advanced Data Augmentation and Optuna-based Hyperparameter Optimization

Trinh Tran Tran, Dang Van Thin
University of Information Technology-VNUHCM,
Vietnam National University, Ho Chi Minh City, Vietnam
23521624@gm.uit.edu.vn
thindv@uit.edu.vn

Abstract

Detecting hate and hope speech in Arabic social media remains a critical challenge in the MAHED 2025 Shared Task (Zaghouani et al., 2025) due to the complex diglossia, diverse dialects, and prevalent orthographic noise in user-generated texts. We introduce a multilingual transformer ensemble that integrates three complementary encoders—AraBERTv2, AraBERT-Twitter, and XLM-RoBERTa—using a uniform soft voting approach (Salur and Aydın, 2022). Each model is fine-tuned with a balanced data augmentation strategy, combining 70% back-translation and 30% Easy Data Augmentation (EDA), followed by noise induction to mimic real-world textual perturbations (Bayer et al., 2022). Hyperparameters are optimized via Optuna (Akiba et al., 2019) to maximize macro-F1 performance. Our method achieves a macro-F1 score of 0.65 on the official test set, surpassing the strongest single model by 0.04 and outperforming competitive multilingual baselines such as mBERT and LLaMA-based Arabic large language models. These results demonstrate that combining complementary linguistic representations with targeted augmentation substantially improves robustness across dialects and addresses class imbalance in Arabic hate and hope speech classification.

1 Introduction

User-generated Arabic text on social media spans *hope* speech—promoting positivity and inclusivity—and *hate* speech—spreading hostility and division. Distinguishing between them is both a computational challenge and a socially impactful task, as online discourse influences public opinion and cohesion.

Arabic presents unique difficulties: *diglossia* between Modern Standard Arabic (MSA) and regional dialects, rich morphology that increases data sparsity, and *orthographic noise* (inconsistent spellings, elongations, and code-switching) that hinders generalization (Darwish et al., 2021).

The MAHED 2025 Shared Task (Sub-task 1) addresses these challenges by providing an imbalanced benchmark (over half *not_applicable*), making macro-F1 (Dalianis, 2018) a more reliable metric than accuracy. Success requires robustness to dialectal variation, noise, and minority-class recall loss.

We propose a **multilingual transformer ensemble** integrating AraBERTv2, AraBERT-Twitter, and XLM-RoBERTa via uniform soft voting. Each model is trained with a balanced augmentation pipeline (70% back-translation, 30% EDA) followed by noise induction, and tuned using Optuna for optimal macro-F1.

Our contributions are:

- A **targeted augmentation pipeline** balancing semantic fidelity and lexical diversity.
- **Optuna-based hyperparameter search** for principled tuning of Arabic-capable transformers.
- A **complementary ensemble** achieving +0.04 macro-F1 over the best single model.

2 Related Work

Hate speech detection has progressed from traditional machine learning with handcrafted features (Schmidt and Wiegand, 2017) to transformer-based models that capture rich contextual representations.

For Arabic, earlier methods using n-grams and sentiment lexicons struggled with complex morphology and dialectal diversity. AraBERT (Antoun et al., 2020) addressed this via morphology-aware tokenization and large-scale Arabic pre-training, while AraBERT-Twitter incorporated social media data to improve handling of informal and dialectal text.

Data augmentation techniques such as back-translation (Taheri et al., 2024) and Easy Data Augmentation (EDA) (Wei and Zou, 2019) have im-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 603–607
November 8-9, 2025 ©2025 Association for Computational Linguistics

proved performance in low-resource, imbalanced scenarios. However, prior Arabic-focused studies typically used them in isolation, without exploring balanced combinations or integration with noise-based perturbations to reflect real-world input conditions.

Multilingual models like mBERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) transfer well to Arabic, but may lack robustness to noisy social media text. Recent Arabic-adapted LLaMA variants achieve competitive results but are resource-intensive.

Ensemble methods (Juola, 2022) improve robustness, yet most Arabic NLP ensembles combine similar models, limiting diversity. Our work differs by combining three complementary transformers—formal MSA, informal/dialectal Twitter, and multilingual—via soft voting, alongside a balanced hybrid augmentation pipeline with noise induction and principled hyperparameter tuning.

3 Background

3.1 Task Setup

Given an Arabic social media post, the task is to predict one of three categories: `hate`, `hope`, or `not_applicable`. The principal evaluation metric is macro-F1, chosen to address class imbalance and linguistic diversity. Importantly, the validation and test labels are hidden from participants. Predictions must be submitted to the official leaderboard to receive macro-F1 scores, fostering robust generalization and precluding tuning on these sets.

3.2 Dataset

The dataset (Zaghouani et al., 2024) comprises posts from multiple platforms, manually annotated by native speakers. Table 1 shows the distribution, with *not_applicable* forming over half of the data, potentially biasing models. Dialects span Egyptian, Gulf, Levantine, and Maghrebi, adding linguistic diversity.

	Train	Dev	Test
Hate	1,301	-	-
Hope	1,892	-	-
Not_applicable	3,697	-	-
Total	6,890	1,476	1,477

Table 1: Dataset statistics.

4 System Overview

Our system combines complementary models, augmentation, and optimization.

4.1 Model Choice

We ensemble three transformers with distinct strengths:

- **AraBERTv2**: strong in MSA morphology and syntax.
- **AraBERT-Twitter**: captures informal, dialectal social media language.
- **XLM-RoBERTa**: handles code-switching and rare tokens via multilingual subword coverage.

4.2 Data Augmentation

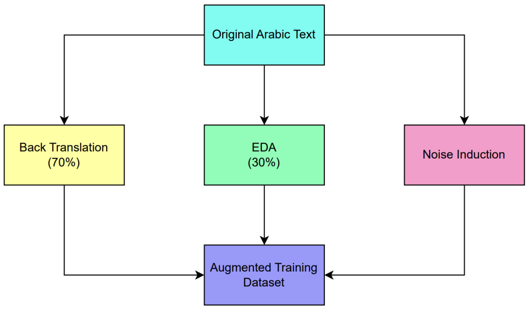


Figure 1: Three-stage data augmentation pipeline.

As shown in Figure 1, the augmentation process begins with the original Arabic text, which is split into two main branches: 70% for **Back Translation** (Arabic→English→French→Arabic) and 30% for **EDA** (synonym replacement, random insertion, swap, deletion). These two branches are then merged and passed through a **Noise Induction** stage, introducing character-level perturbations to mimic real-world orthographic errors. This design intentionally balances semantic fidelity (from BT) with lexical diversity (from EDA), while Noise Induction strengthens robustness to typos, elongations, and informal spellings that are frequent in social media data. Empirically, this configuration achieved the best macro-F1 on the development set compared to using any single augmentation method alone.

4.3 Ensemble Strategy

Figure 2 illustrates the final ensemble architecture. It integrates **AraBERTv2** (specialized in

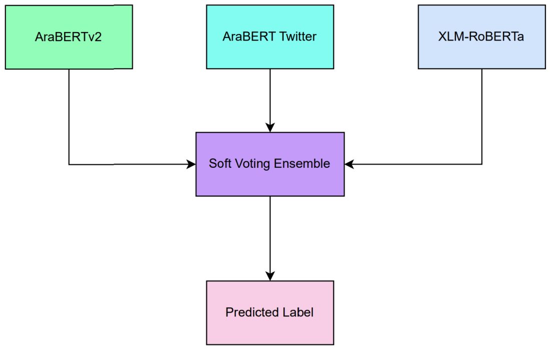


Figure 2: Soft-voting ensemble combining three complementary transformer models.

MSA), **AraBERT-Twitter** (optimized for informal/dialectal text), and **XLM-RoBERTa** (multilingual with strong cross-lingual transfer). We apply **uniform soft voting**, where the predicted probabilities from each model are averaged before selecting the label with the highest mean score. This method exploits complementary strengths—MSA precision, dialect coverage, and code-switch handling—while avoiding over-reliance on a single model. Notably, soft voting preserves high-confidence predictions for minority classes like *hope*, boosting recall without harming overall accuracy.

4.4 Preprocessing

Normalization includes: diacritic removal, Alef normalization, elongation stripping, and removal of non-Arabic symbols/emojis, improving token consistency.

4.5 Hyperparameter Optimization

Optuna tunes learning rate, batch size, weight decay, and dropout over 20 trials, optimizing macro-F1 with early stopping.

4.6 Tools

Implemented in PyTorch 2.2 + HuggingFace Transformers 4.39, trained on Kaggle P100 GPUs with public checkpoints for reproducibility. All models and hyperparameter tuning are performed solely on the training set, following the competition protocol that prohibits using validation or test labels for training or tuning. Evaluation on validation and test sets is conducted via blind leaderboard submissions.

5 Results

5.1 Main Results

Table 2 presents the macro-F1 scores on the MAHED 2025 test set. Among single models, **AraBERT-Twitter** achieves the highest score (0.61), benefiting from its pre-training on informal, dialectal Arabic that closely matches the dataset's social media origin. AraBERTv2 and XLM-RoBERTa follow closely (0.60 each), with the former excelling in MSA-heavy samples and the latter leveraging cross-lingual patterns to handle code-switching and rare dialectal tokens.

Our **soft-voting ensemble** (Figure 2) achieves a macro-F1 of **0.65**, a +0.04 absolute improvement over the strongest single model. In highly imbalanced, noisy classification settings like MAHED 2025, such gains indicate a substantive boost in **robustness** and **dialectal coverage**. The improvement predominantly comes from higher recall in the minority *hope* class while maintaining precision for *hate* and *not_applicable*. This effect is consistent with the design in Figure 2: soft voting allows confident minority-class predictions from one model to be preserved, even when two other models disagree, preventing majority-class dominance.

The ensemble's performance gain is attributable to three complementary competencies:

- **MSA precision** from AraBERTv2.
- **Dialect sensitivity** from AraBERT-Twitter.
- **Cross-lingual generalization** from XLM-RoBERTa.

Because validation and test labels are withheld, we rely on the leaderboard feedback for validation performance. Final test set results reflect true generalization under realistic blind test conditions.

Model	Macro-F1
AraBERTv2	0.60
AraBERT Twitter	0.61
XLM-RoBERTa	0.60
Ensemble	**0.65**

Table 2: Test set performance of individual models and our ensemble.

5.2 Ablation Study

To isolate the contribution of each augmentation component in the pipeline shown in Figure 1, we conducted controlled experiments with different augmentation settings (Table 3).

When applied individually, **EDA** and **Back Translation (BT)** provide only marginal gains over the no-augmentation baseline (+0.01 to +0.02 macro-F1). **Noise Induction** alone yields negligible benefit, suggesting that robustness to orthographic noise must be paired with semantic or lexical diversity to be effective.

The **full pipeline**—70% BT, 30% EDA, plus noise induction on all augmented samples—achieves the highest macro-F1 of **0.65**. This aligns with the design rationale in Figure 1:

- BT preserves semantic fidelity while generating dialectal and syntactic variations.
- EDA injects controlled lexical and word-order diversity, enabling better generalization.
- Noise Induction trains the model to withstand character-level perturbations common in social media.

Compared to the baseline (0.62), the combined approach delivers a +0.03 absolute gain, directly enabling the ensemble's boost reported in Table 2.

Augmentation	Macro-F1
No Augmentation	0.62
EDA only	0.59
Back Translation only	0.60
Noise Induction only	0.59
BT + EDA + Noise Induction	**0.65**

Table 3: Macro-F1 results for different augmentation settings.

6 Conclusion

We presented a multilingual transformer ensemble for the MAHED 2025 hate and hope speech classification task, targeting one of the most challenging scenarios in Arabic NLP: diglossia, dialectal variation, and noisy user-generated text. Our approach combines three complementary encoders—AraBERTv2, AraBERT-Twitter, and XLM-RoBERTa—through a uniform soft-voting strategy, each fine-tuned with a carefully balanced data augmentation pipeline (70% back-translation, 30% EDA, plus noise induction). Hyperparameters were optimized using Optuna, enabling the models to adapt to the dataset's imbalance and orthographic variability.

The system achieves a **macro-F1 of 0.65** on the official test set, outperforming the strongest single model by +0.04 absolute and surpassing competitive multilingual baselines such as mBERT and Arabic LLaMA derivatives. Our ablation analysis confirms that augmentation diversity and model complementarity are key to robust performance, especially in the minority *hope* class.

Practical Implications: Beyond the shared task, our findings suggest that: (i) balanced multi-technique augmentation can outperform single-method augmentation in low-resource, imbalanced, and noisy settings; (ii) soft-voting ensembles mitigate individual model biases without requiring heavy training of meta-classifiers; and (iii) robustness to orthographic noise is not optional—it is critical for social media Arabic.

Future Work: We plan to explore: (a) adaptive ensemble weighting learned from development set meta-features; (b) integration of large language model embeddings for richer semantic context; (c) domain adaptation to handle sarcasm, figurative speech, and evolving slang; and (d) multi-modal fusion with images and metadata to capture context beyond text.

Limitations: Our back-translation process depends on third-party APIs, which may introduce domain bias. We also did not conduct statistical significance testing to quantify the reliability of observed improvements. Finally, while our augmentation pipeline is effective, it is computationally more expensive than single-method augmentation, which could be a constraint in real-time systems.

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

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 & data mining*, pages 2623–2631.

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 Process-*

ing Tools, with a Shared Task on Offensive Language Detection, pages 9–15, Marseille, France. European Language Resource Association.

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Hercules Dalianis. 2018. Evaluation metrics and evaluation. In *Clinical Text Mining: secondary use of electronic patient records*, pages 45–53. Springer.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R El-Beltagy, and Wassim El-Hajj. 2021. A panoramic survey of natural language processing in the arab world. *Communications of the ACM*, 64(4):72–81.

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.

Patrick Juola. 2022. Ensemble methods. In *Encyclopedia of Big Data*, pages 437–438. Springer.

Mehmet Umut Salur and İlhan Aydın. 2022. A soft voting ensemble learning-based approach for multimodal sentiment analysis. *Neural Computing and Applications*, 34(21):18391–18406.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Alireza Taheri, Azadeh Zamanifar, and Amirfarhad Farhadi. 2024. Enhancing aspect-based sentiment analysis using data augmentation based on back-translation. *International Journal of Data Science and Analytics*, pages 1–26.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

A Appendix

Full hyperparameters and code are available at: `https://github.com/trantranuit/mahed2025-system`.

YassirEA at MAHED 2025: Fusion-Based Multimodal Models for Arabic Hate Meme Detection

Yassir El Attar
Institute of Natural Language Processing, University of Stuttgart
yassir.el.attar@gmail.com

Abstract

We present our system for the MAHED 2025 Shared Task on Arabic Hate Meme Detection (subtask 3), a binary classification task to determine whether a multimodal meme containing Arabic text and an image conveys a hateful message. Our approach uses multimodal fusion combining a visual encoder and an Arabic text encoder. We explored four fusion strategies—transformer fusion, early fusion, cross-attention, and bilinear fusion—and found transformer fusion offered the best single-model trade-off, while an ensemble of all four achieved the highest score. To address the severe class imbalance (90.05% not-hate vs. 9.95% hate), we applied class-weighted loss, focal loss, strong regularization, and light augmentation. Our best submission reached a macro-F1 score of **0.75** on the gold test set.

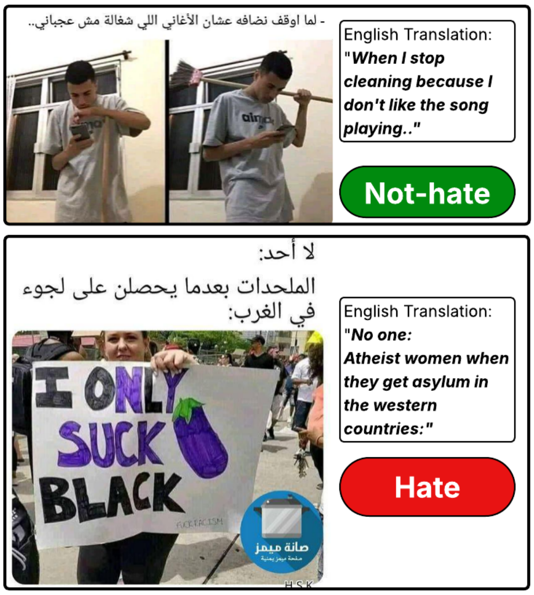

- لما اوقف نضافه عشان الأغاني اللي شغالة مش عجباني..
English Translation:
"When I stop cleaning because I don't like the song playing.."
Not-hate
لا أحد:
الملحدات بعدما يحصلن على لجوء
في الغرب:
I ONLY SUCK BLACK
FUCK RACISM
صانة ميمز
صفحة ميمز يمنية
English Translation:
"No one:
Atheist women when they get asylum in the western countries:"
Hate

Figure 1: Examples of *hate*/*not-hate* memes from the **Evaluation-phase test split**.

1 Introduction

Social media enables rapid information sharing but also accelerates the spread of harmful content, including hate speech. While text-only hate speech detection is well studied, much hateful content now appears in **multimodal formats**, such as memes, which combine text and images into a single communicative unit. These memes often use humor, irony, or cultural symbols to mask or amplify harmful messages, making automated detection challenging (Kiela et al., 2021; Boishakhi et al., 2021). Figure 1 shows examples of Arabic memes from the two classes (*hate* and *not-hate*), illustrating the diversity in visual style and text content.

The **MAHED 2025 Shared Task** (Zaghouani et al., 2025) targets hateful meme detection in Arabic, a language with rich morphology, diverse dialects, and high orthographic variation. Memes may contain Modern Standard Arabic, dialectal Arabic, or a mix, with images referencing culturally specific or political contexts (Mubarak et al., 2023). These factors, along with OCR errors, slang, and stylized fonts, complicate feature extraction. Modeling the interplay between Arabic text and images requires fine-grained cross-modal alignment, motivating our exploration of multiple multimodal fusion strategies.

We address the task under two constraints: a small, imbalanced dataset and the need for effective multimodal fusion. Using state-of-the-art encoders for text and vision, we compare four fusion mechanisms and evaluate an ensemble.

This work makes three main contributions:

1. We provide a systematic comparison of four fusion strategies for Arabic multimodal hate detection.

2. We conduct an in-depth analysis of strategies to mitigate extreme class imbalance, including class-weighted loss, focal loss, and multimodal augmentation.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 608–614
November 8-9, 2025 ©2025 Association for Computational Linguistics

3. We release a public, reproducible system design[1] that can serve as a baseline for future Arabic multimodal classification tasks.

2 Background

Detecting hate speech in multimodal content has become a major research area, especially following the release of the *Hateful Memes* benchmark, which exposed the limitations of unimodal systems in handling cross-modal semantics (Kiela et al., 2021). Subsequent work has explored a range of fusion techniques, including early fusion (concatenating text and image embeddings before classification) (Galanakis et al., 2025), late fusion (combining predictions from unimodal models) (Snoek et al., 2005), and intermediate, attention-based approaches such as cross-attention and co-attention (Lu et al., 2017, 2019; Chen et al., 2020; Zhang et al., 2024).

In Arabic NLP, hate speech detection has mostly focused on text-only methods (Mubarak et al., 2023; Al-Saqqa et al., 2024) using pretrained language models such as AraBERT (Antoun et al., 2020), CAMeLBERT (Inoue et al., 2021), and MARBERTv2 (Abdul-Mageed et al., 2021). Vision–language pretraining models such as CLIP (Radford et al., 2021), SigLIP (Zhai et al., 2023), and Swin Transformer (Wang and Markov, 2024) have also shown promise for multimodal classification. However, their effectiveness for Arabic multimodal hate detection remains underexplored.

3 System Overview

In preliminary experiments on the development set, we found that combining MARBERTv2 for text with CLIP-Large for images performed best. Our final system is therefore built on this pairing, with the overall architecture described in Section 3.1. We also experimented with a uni-modal approach where each modality is used separately for the predictions (details can be found in Appendix E)

3.1 Model Components

Figure 2 illustrates the overall architecture of our system. The input meme consists of an image and its corresponding Arabic text. The image is processed by a visual encoder (CLIP-Large), producing image embeddings, while the text is processed by an Arabic text encoder (MARBERTv2) to produce text embeddings. These embeddings are then fed into one of four fusion mechanisms—transformer fusion, early concatenation, cross-attention, and bilinear pooling—which learn joint multimodal representations. The outputs of all fusion models are combined in an ensemble module that produces the final prediction as either *hate* or *not-hate*.

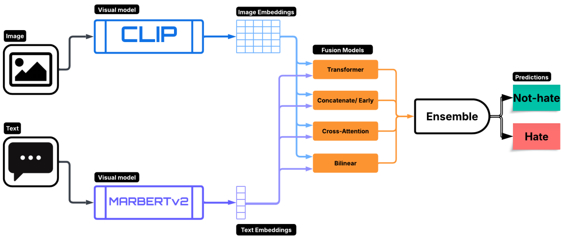


Figure 2: Framework overview

Unimodal Representations. We process the meme text[2] using **MARBERTv2**, a transformer-based language model pretrained on large-scale Arabic text from social media. We take the final hidden state of the `[CLS]` token as the text embedding.

For the image, we use **CLIP-Large (ViT-L/14)** (Radford et al., 2021) to generate visual features. We take the pooled output from CLIP's image encoder as the image embedding.

3.2 Fusion Mechanisms

We explore four fusion strategies, all of which fall under early or intermediate fusion: the text embedding $\mathbf{t} \in \mathbb{R}^{d_t}$ from the text encoder and the image embedding $\mathbf{v} \in \mathbb{R}^{d_v}$ from the vision encoder are merged into a joint representation.

Concatenation (Early Fusion). The text and image embeddings are concatenated into a single vector and passed through a feed-forward layer with ReLU activation and dropout before classification; see Eq. (1) in Appendix C. Here, $[\mathbf{t}; \mathbf{v}]$ denotes concatenation, W and W_o are weight matrices, and $\mathbf{b}$ and $\mathbf{b}_o$ are biases.

Transformer Fusion (Single-Stream). A lightweight transformer jointly processes projected text ($\mathbf{t}$) and image ($\mathbf{v}$) embeddings of equal dimension d, augmented with modality type embeddings. The two-token sequence passes

[1] https://github.com/YassirELATTAR/task3-mahed2025

[2] The extracted text was provided as part of the task data.

through L self-attention layers, and the pooled token is classified with a small MLP.[3]

Cross-Attention (Dual-Stream). Two single-head cross-attention blocks let text attend to image features and vice versa, aligning modalities more explicitly than concatenation but typically requiring more data to generalize.

Bilinear Fusion. Multimodal Compact Bilinear (MCB) pooling (Fukui et al., 2016) models multiplicative interactions between $\mathbf{t}$ and $\mathbf{v}$ in a compressed space, enabling richer feature combinations at the cost of higher overfitting risk on small datasets.

3.2.1 Ensemble

We combine the predictions of all fusion models using:

- **Majority Vote:** Label predicted by most models.
- **Equal Weighted:** Mean-pooling of class probabilities before selecting the argmax.
- **Transformer-Weighted:** Weighted average giving higher weight to transformer fusion[4].

3.3 Dealing with Imbalance

A major challenge in this task is the severe class imbalance in the training data (90.05% *not-hate* vs. 9.95% *hate*). To address this, we experimented with several training-time strategies.

Class-Weighted Training Loss. We use weighted cross-entropy with inverse-frequency class weights; see Eq. (2) in Appendix C. This increases the penalty for errors on the minority class.

Focal Loss. We also test focal loss (Lin et al., 2018) to focus more on hard examples (Eq. (3) in Appendix C), where γ controls hard-example emphasis and α is set to the minority-class prior.

Regularization. To reduce overfitting to the majority class, we applied stronger dropout (0.3 in encoders, 0.2 in fusion layers), weight decay (10^{-4}), and early stopping (patience 5).

Targeted Data Augmentation. To balance the dataset, we augmented the *hate* class with both modified images and texts. For images, we applied rotation, scaling, perspective warp, color jitter, gamma adjustment, noise/blur, geometric distortions, shadows/fog, and crop–resize. For text, we used OCR-extracted text from augmented images (70% probability when confidence was high), synonym replacement, light character dropout, and cautious AR→EN→AR back-translation. We designed the augmentation to preserve the original semantic intent. We paired augmented images and text in three ways: (i) replacing the text with the newly extracted text, (ii) appending new text to the original, and (iii) substituting a few words without altering the meaning. (We show a few examples in Appendix B.)

4 Experimental Setup

4.1 Data and Evaluation

The task is to determine whether a multimodal meme—comprising an image and embedded Arabic text—conveys a hateful message (*hate*) or not (*not hate*). This phenomenon often involves *meaning multiplication*: even if neither the text nor the image alone is hateful, their combination can create a hateful meaning. Effective fusion of the two modalities is therefore crucial, and in this work we explore different fusion strategies.

We use the official splits from the Prop2Hate-Meme dataset (Alam et al., 2024b,a), which follow the shared task protocol for training, development, and testing. The training split is highly imbalanced, with 90.05% *not-hate* and only 9.95% *hate* examples. This motivates the imbalance-handling strategies described in Section 3.3. No external labeled data are used. The official evaluation metric for the shared task, and for all our experiments, is **Macro-F1**, which is preferred over accuracy because it balances performance across classes in the presence of severe class imbalance.

4.2 Training and Evaluation

We trained the following models in our experiments:

- **MARBERTv2** (Abdul-Mageed et al., 2021) as the Arabic text encoder[5].
- **CLIP-Large (ViT-L/14)** (Radford et al., 2021) as the visual encoder[6].

[3] This was the strongest single-model method in preliminary validation.

[4] This choice is based on its stronger validation performance compared to other models.

[5] https://huggingface.co/UBC-NLP/MARBERTv2

[6] https://huggingface.co/openai/clip-vit-large-patch14

Fusion	Accuracy	Macro-F1 (Test)	Macro-F1 (Gold)
Ensemble (All)	0.90	0.72	**0.75**
Transformer	0.91	0.72	0.75
Concatenation	0.89	0.74	0.73
Cross-Attn.	0.88	0.69	0.68
Bilinear	0.89	0.63	0.66

Table 1: Performance on evaluation-phase test (Test) and official leaderboard (Gold) splits. Ensemble gain over Transformer = +0.005 on Gold.

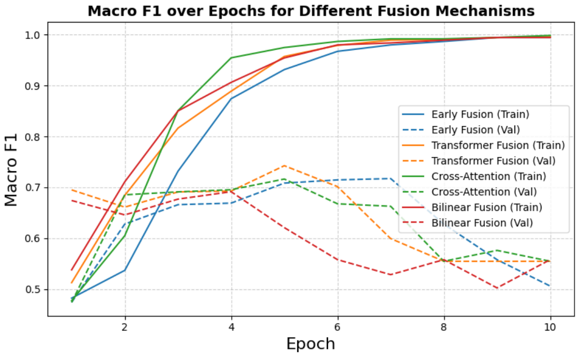


Figure 3: Macro-F1 progression across epochs on Train (solid) and Development (dotted). *Takeaway:* Transformer fusion is the most stable and highest-performing; bilinear overfits quickly.

- Four fusion architectures: concatenation (early fusion), transformer fusion, cross-attention (dual-stream), and bilinear fusion.
- An ensemble combining the predictions of all four fusion models.

We trained all models on the official train split and tuned them on the development set, using **Macro-F1** as the model selection criterion. Details of the hyperparameters are reported in Appendix D.

5 Results

Table 1 presents the main results on MAHED Subtask 3. We report **Macro-F1** on the test split provided during the evaluation phase, and **Macro-F1*** on the gold test set. The latter corresponds to the official leaderboard score. Macro-F1 is the primary evaluation metric of the shared task because it balances performance across classes in the presence of severe class imbalance (Section 3.3).

Figure 3 visualizes the progression of Macro-F1 over training epochs for each fusion mechanism on the train and dev splits. To better illustrate the overfitting behaviour, we train for 10 epochs without early stopping, while keeping all other hyperparameters the same.

Observations. Transformer fusion offers the best single-model trade-off between capacity and stability. The ensemble slightly improves Macro-F1 (+0.005 on the test split) but at the cost of a small drop in accuracy. Cross-attention underperforms transformer fusion, likely due to limited training data, while bilinear fusion tends to overfit. For imbalance handling, class-weighted loss yields the most consistent improvements. Focal loss reduces the impact of easy majority-class cases and can slightly improve minority recall, but the gain is marginal. Data augmentation does not improve performance—in fact, the model often overfits to the augmented data, reaching perfect scores on the training set but dropping significantly on dev. A possible explanation is that the augmented samples introduce superficial patterns that the model can exploit without learning meaningful cross-modal interactions.

Example predictions for the two samples shown in Figure 1 are provided in Appendix A.

6 Limitations

Our system depends on pre-extracted texts from memes, which may miss stylized text; sarcasm/irony and culture-specific references remain challenging. The dataset's class imbalance and limited size constrain generalization, with bilinear and cross-attention models prone to overfitting. We did not perform Arabic-specific vision–language pretraining, which could improve alignment.

7 Conclusion

We explored different fusion strategies combining an Arabic text encoder and a visual encoder for Arabic hate meme detection. We find that an ensemble that aggregates the individual predictions is most effective, yielding a Macro-F1 score of **0.75** on the official test set and ranking **second** on the shared task leaderboard. We also examined approaches to mitigate class imbalance, including class-weighted loss, focal loss, and regularization, and find class-weighted loss to be the most effective. Future work could investigate culture-aware prompts and Arabic-focused vision–language pretraining. Our findings can guide the development of future Arabic multimodal hate detection systems.

Acknowledgments

Thanks are extended to the MAHED 2025 organizers and the dataset providers. A heartfelt thanks also to Prof. Dr. Carina Silberer for her guidance and assistance and to the Institute of Natural Language Processing (IMS), University of Stuttgart, for providing the working environment and GPU server resources used in this work.

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.

Samar Al-Saqqa, Arafat Awajan, and Bassam Hammo. 2024. A survey of hate speech detection for arabic social media: Methods and datasets. *Procedia Computer Science*, 251:224–231.

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmad, Md. Arid Hasan, and Maram Hasanain. 2024b. ArMeme: Propagandistic content in Arabic memes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21071–21090, Miami, Florida, USA. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of the LREC*.

Fariha Tahosin Boishakhi, Ponkoj Chandra Shill, and Md. Golam Rabiul Alam. 2021. Multi-modal hate speech detection using machine learning. In *2021 IEEE International Conference on Big Data (Big Data)*, page 4496–4499. IEEE.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. *Preprint*, arXiv:1909.11740.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, Austin, Texas. Association for Computational Linguistics.

Ioannis Galanakis, Rigas Filippos Soldatos, Nikitas Karanikolas, Athanasios Voulodimos, Ioannis Voyiatzis, and Maria Samarakou. 2025. Early and late fusion for multimodal aggression prediction in dementia patients: A comparative analysis. *Applied Sciences*, 15(11).

Go Inoue, Muhammad Abdul-Mageed, and Mahmoud El-Haj. 2021. Camelbert: Transformer-based arabic language models. In *Proceedings of WANLP*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2021. The hateful memes challenge: Detecting hate speech in multimodal memes. *Preprint*, arXiv:2005.04790.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *Preprint*, arXiv:1708.02002.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Preprint*, arXiv:1908.02265.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2017. Hierarchical question-image co-attention for visual question answering. *Preprint*, arXiv:1606.00061.

Hamdy Mubarak, Sabit Hassan, and Shammur Absar Chowdhury. 2023. Emojis as anchors to detect arabic offensive language and hate speech. *Natural Language Engineering*, 29(6):1436–1457.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Cees Snoek, Marcel Worring, and Arnold Smeulders. 2005. Early versus late fusion in semantic video analysis. pages 399–402.

Yeshan Wang and Ilia Markov. 2024. CLTL@multimodal hate speech event detection 2024: The winning approach to detecting multimodal hate speech and its targets. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 73–78, St. Julians, Malta. Association for Computational Linguistics.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. Overview of the mahed shared task: Multimodal detection of hope and hate emotions in arabic content. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.

Yinghui Zhang, Tailin Chen, Yuchen Zhang, and Zeyu Fu. 2024. Enhanced multimodal hate video detection via channel-wise and modality-wise fusion. In *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, page 183–190. IEEE.

Appendix

A Example Predictions

Table 2 shows the predictions from different fusion models for the two examples in Figure 1.

Model	Example 1	Example 2
(Ground truth)	not-hate	hate
Concatenation	not-hate	not-hate
Transformer	not-hate	hate
Cross-Attn.	not-hate	hate
Bilinear	not-hate	hate
Ensemble	not-hate	hate

Table 2: Example predictions from different models.

B Augmentation Examples

Figure 4 illustrates examples of image augmentations applied to the *hate* class. The associated text augmentations are shown below each image.

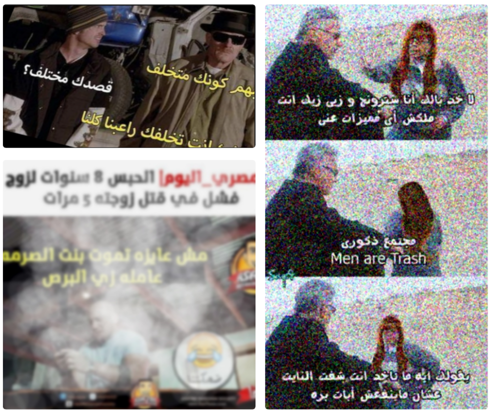


Figure 4: Examples of image augmentations for the *hate* class.

C Additional Modeling Equations

Concatenation (early fusion).

$$\mathbf{h} = \mathrm{ReLU}\big(W[\mathbf{t};\mathbf{v}] + \mathbf{b}\big),$$
$$\hat{\mathbf{y}} = \mathrm{softmax}\big(W_o\ \mathrm{Dropout}(\mathbf{h}) + \mathbf{b}_o\big). \quad (1)$$

Table 3: Baseline summary on the test set (accuracy and macro F1).

Approach	Acc (Weighted)	Macro-F1 (Weighted)	Acc (Focal)	Macro-F1 (Focal)
Text only	0.80	0.67	0.76	0.57
Image only	0.77	0.57	0.77	0.59
Confidence combine	0.78	0.59	0.78	0.55

Table 4: Text-only (Weighted) – classification report (test).

Class	Precision	Recall	F1	Support
not-hate	0.81	0.96	0.88	452
hate	0.74	0.34	0.46	154
Accuracy		0.80		606
Macro avg	0.78	0.65	0.67	606
Weighted avg	0.79	0.80	0.77	606

Weighted cross-entropy.

$$\mathcal{L}_{\mathrm{wCE}} = -\,w_1\,y\log p \;-\; w_0\,(1-y)\,\log(1-p). \quad (2)$$

Focal loss.

$$\mathcal{L}_{\mathrm{focal}} = -\,\alpha\,(1-p)^{\gamma}\,y\log p$$
$$-\;(1-\alpha)\,p^{\gamma}\,(1-y)\,\log(1-p). \quad (3)$$

D Framework Training Details and Hyperparameters

The main hyperparameters used: batch size 16, 40 training epochs, AdamW optimizer, base learning rate 2×10^{-5} with a linear scheduler, and weight decay of 10^{-4}. We applied dropout of 0.3 in encoders and 0.2 in fusion layers, and used early stopping with patience 5 to prevent overfitting.

E Unimodal Experiments

We evaluate three simple baselines: (i) text only, (ii) image only, and (iii) a confidence-based combination of the two unimodal systems (if the two disagree, pick the class from the model with higher softmax confidence). Each is trained/evaluated under class-weighted cross-entropy and Focal Loss. We report test accuracy and macro F1, then the final classification reports.

Table 5: Text-only (Focal) – classification report (test).

Class	Precision	Recall	F1	Support
not-hate	0.77	0.96	0.86	452
hate	0.60	0.18	0.28	154
Accuracy		0.76		606
Macro avg	0.69	0.57	0.57	606
Weighted avg	0.73	0.76	0.71	606

Table 6: Image-only (Weighted) – classification report (test).

Class	Precision	Recall	F1	Support
not-hate	0.78	0.97	0.86	452
hate	0.66	0.18	0.28	154
Accuracy		0.77		606
Macro avg	0.72	0.57	0.57	606
Weighted avg	0.75	0.77	0.71	606

Table 7: Image-only (Focal) – classification report (test).

Class	Precision	Recall	F1	Support
not-hate	0.78	0.97	0.87	452
hate	0.70	0.19	0.30	154
Accuracy		0.77		606
Macro avg	0.74	0.58	0.58	606
Weighted avg	0.76	0.77	0.72	606

Table 8: Confidence-based combination (Weighted) – classification report (test).

Class	Precision	Recall	F1	Support
not-hate	0.78	0.98	0.87	452
hate	0.76	0.19	0.30	154
Accuracy		0.78		606
Macro avg	0.77	0.58	0.59	606
Weighted avg	0.78	0.78	0.72	606

Table 9: Confidence-based combination (Focal) – classification report (test).

Class	Precision	Recall	F1	Support
not-hate	0.77	1.00	0.87	452
hate	0.91	0.14	0.24	154
Accuracy		0.78		606
Macro avg	0.84	0.57	0.55	606
Weighted avg	0.81	0.78	0.71	606

AAA at MAHED Shared task: A Systematic Encoder Evaluation for Arabic Hope and Hate Speech Classification

Ahmed Elzainy, Hazem Abdelsalam, Ahmed Samir[*]**,**
Mohamed Amin[*]
Alexandria university, Egypt
{es-ahmedkhalil2025, es-Hazemabdelsalam2024, es-AhmedAbdelMaksoud2025,
es-MohamedE.Amin2025}@alexu.edu.eg

Abstract

Arabic hate speech detection presents unique challenges due to the language's morphological complexity, dialectal diversity, and the subtle nature of emotional expressions in social media. In this paper, we present our submission to the MAHED shared task for Arabic hate speech classification, which aims to classify Arabic text into three categories: hope, hate, and not_applicable. This task is crucial for building safer online communities and has applications in content moderation, social media analysis, and digital wellbeing initiatives. We systematically evaluate six transformer-based encoders, comparing Arabic-specific models (MARBERT, AraBERT, ALCALM) against multilingual alternatives (XLM-RoBERTa, LaBSE, BGE). Our approach demonstrates that specialized Arabic models specially encoders trained on more than one dialect like marber significantly outperform their multilingual counterparts, with MARBERT achieving the best overall performance. Using our proposed methodology, we achieved competitive results on the MAHED shared task with a macro-F1 score of 0.707 on the test split, securing a strong position in the final competition rankings.

1 Introduction

This paper details the system we developed for the MAHED (Multimodal Detection of Hope and Hate Emotions in Arabic Content) shared task, hosted at the Arabic Natural Language Processing Conference (ArabicNLP 2025)(Zaghouani et al., 2025). Our work addresses the critical challenge of Arabic hate speech classification, a multi-class problem designed to distinguish between hope, hate, and neutral expressions in Arabic social media content.

The importance of this task has grown substantially with the increasing prevalence of Arabic content online and the urgent need for effective content moderation systems. Robust hate speech detection systems have critical real-world applications in social media platforms for automated content filtering, in digital wellbeing initiatives for protecting vulnerable users, in research contexts for understanding online discourse patterns, and in policymaking for developing evidence-based regulations around online hate speech.

The challenge of emotion classification in Arabic is particularly acute due to the language's intrinsic complexities. Arabic is characterized by its rich morphological system, where words can be derived from trilateral or quadrilateral roots through complex patterns, making surface-level features less reliable. Furthermore, the phenomenon of diglossia—the coexistence of Modern Standard Arabic (MSA) with numerous regional dialects—means that emotional expressions often carry dialectal nuances that may not be immediately apparent to standard language models. Additionally, the subtlety of hate speech and sarcastic expressions in Arabic social media creates challenges for automated detection systems that must distinguish between explicit and implicit emotional content.

To address these challenges, we conducted a systematic evaluation of six transformer-based encoders, comparing their effectiveness on Arabic emotion classification. Our methodology focuses on fine-tuning individual models with careful hyperparameter optimization rather than ensemble approaches, allowing us to identify the most capable single-model solution for deployment scenarios where computational efficiency is crucial.

The key contributions and findings of our work can be summarized as follows:

- We demonstrate the systematic evaluation of six diverse transformer encoders on Arabic emotion classification, providing comprehensive performance comparisons across Arabic-specific and multilingual models.

[*] Equal contribution

Proceedings of The Third Arabic Natural Language Processing Conference, pages 615–619
November 8-9, 2025 ©2025 Association for Computational Linguistics

- We show that Arabic-specific models (MARBERT(Abdul-Mageed et al., 2021), ALCLAM(Murtadha et al., 2024), AraBERT(Antoun et al.)) significantly outperform multilingual alternatives, with MARBERT achieving the best balance of performance and robustness.

2 Background

Arabic hate speech detection has evolved from traditional machine learning approaches using handcrafted features to modern transformer-based methods. Early work in Arabic sentiment analysis relied on lexicon-based approaches and statistical features, but these methods struggled with the morphological richness and dialectal variation of Arabic text.

The introduction of pre-trained language models revolutionized Arabic NLP, with BERT-based models like AraBERT (Antoun et al.) and MARBERT (Abdul-Mageed et al., 2021) demonstrating significant improvements over previous approaches. These models leverage large-scale Arabic corpora to learn contextual representations that better capture the nuances of Arabic text.

Multilingual models such as XLM-RoBERTa (Conneau et al., 2019) have shown competitive performance across multiple languages, but their effectiveness on Arabic-specific tasks remains a subject of investigation. Recent work has suggested that language-specific pre-training often provides advantages for morphologically rich languages like Arabic (Abdul-Mageed et al., 2021)(Antoun et al.).

In our setup, both the input and output are text: the input is a sentence and the output is a label from the set {hope, hate, not_applicable}. For example, the input ابتهاج محمد اول امريكيه تشارك في الاوليمبياد بحجاب ، بدأت النهارده is classified as hope. We evaluate on a dataset of Arabic social media posts (Zaghouani et al., 2025) (train: 6890, validation: 1476, test: 1477), which provides a realistic benchmark for emotion and hate speech detection. This task goes beyond sentiment analysis by requiring fine-grained distinctions between emotional states while handling the informal nature of online discourse.

3 System Overview

3.1 Model Architecture

Our approach employs a standard fine-tuning methodology using transformer-based encoders with task-specific classification heads. The general architecture consists of four main components:

1. **Input Processing**: Text tokenization using model-specific tokenizers optimized for Arabic text handling.
2. **Encoder Layer**: Pre-trained transformer encoder providing contextualized representations of input sequences.
3. **Classification Head**: Linear transformation layer mapping encoder outputs to class probabilities.
4. **Loss Function**: Cross-entropy loss with class weighting to address dataset imbalance.

3.2 Evaluated Models

We systematically evaluate six transformer-based encoders representing different pre-training approaches and language coverage:

Arabic-Specific Models:

- **MARBERT** (Abdul-Mageed et al., 2021): Bidirectional encoder pre-trained specifically on Arabic social media content, optimized for informal Arabic text processing.
- **AraBERT-Twitter** (Antoun et al.): Large-scale Arabic BERT model with Twitter-specific pre-training, designed for social media content understanding.
- **ALCLAM** (Murtadha et al., 2024): Contemplative language model designed for deeper Arabic text understanding and reasoning tasks.

Multilingual Models:

- **XLM-RoBERTa** (Conneau et al., 2019): Cross-lingual encoder supporting 100+ languages, including Arabic, trained on diverse multilingual corpora.
- **LaBSE** (Feng et al., 2022): Language-agnostic sentence encoder designed for cross-lingual text representation and similarity tasks.

- **BGE-m3** (Chen et al., 2024): Bidirectional and generative encoder optimized for text embedding and representation learning.

4 Experimental Setup

4.1 Data Split

	Train	Validation	Test
Number of samples	6890	1476	1477

Table 1: Dataset split for Arabic social media posts.

4.2 Training Setup

Key training parameters for our best-performing model (MARBERT) include:

- Learning rate: 1×10^{-6} (optimized through systematic search)
- Batch size: 64 (training), 128 (evaluation)
- Training epochs: 3 with early stopping
- Warmup ratio: 0.1 for learning rate scheduling
- Weight decay: 0.01 for regularization
- Maximum sequence length: 170 tokens

4.3 Evaluation Framework

Following the shared task guidelines, we employ comprehensive evaluation metrics:

- **Primary Metric**: Macro-averaged F1 score for balanced evaluation across all classes
- **Secondary Metrics**: Accuracy, macro-averaged precision, and recall

5 Results

5.1 Overall Performance

Table 2 presents comprehensive performance comparisons for all models evaluated in the test set. The results demonstrate clear performance advantages for Arabic-specific models over their multilingual counterparts other than alclam model, which we were surprised with the performance of the model.

5.2 Key Findings

The experimental results reveal several important insights:

Model	F1-M	Acc.	Prec.	Rec.
XLM-RoBERTa	0.645	0.653	0.645	0.647
MARBERT	**0.707**	**0.712**	**0.705**	**0.710**
AraBERT-Twitter	0.630	0.658	0.669	0.616
BGE	0.588	0.617	0.631	0.585
ALCLAM Base v2	0.404	0.563	0.684	0.442
LaBSE	0.627	0.654	0.655	0.611

Table 2: Performance comparison across evaluated models on the test set.

Arabic-Specific Model Superiority: ALCALM, MARBERT, and AraBERT-Twitter achieved the highest performance scores, demonstrating the critical importance of language-specific pre-training for Arabic emotion classification tasks.

Multilingual Model Limitations: Bge-m3 showed substantially lower performance despite its broad language coverage, suggesting that multilingual models may not effectively capture the subtle linguistic nuances required for Arabic emotion classification, also XLM-RoBERTa, although higher than both ALCLAM and AraBERT-Twitter, still lower than the MARBERT which is trained on more than one arabic dialect.

Performance-Robustness Trade-offs: MARBERT achieved the best balance across all evaluation metrics, making it the most reliable choice for deployment scenarios requiring consistent performance.

5.3 Error Analysis

Detailed analysis of model predictions on the validation set reveals several patterns in classification errors:

1. **Implicit Hate Expression**: Subtle hate speech often misclassified as neutral content. Example: مايفعله ميسى فى الملعب شكله اخذ منشطاات لازم يفحصونه (What Messi does on the field looks like he took stimulants They need to test him) True: not_applicable, but contains subtle accusatory language that could be misinterpreted.

2. **Dialectal Variation Impact**: Regional dialects and informal expressions create clas-

sification challenges. Example: كسم امتحان انهارده (Damn today's exam) True: not_applicable, but vulgar dialectal expressions can be difficult to classify accurately.

3. **Context-Dependent Statements**: Expressions requiring broader context for accurate interpretation. Example: هذا يقطع الطريق على عصاباتهم المتحالفة معهم (This cuts the road on their allied gangs) True: not_applicable, but political references require contextual understanding for proper classification.

6 Conclusion

The experimental results provide valuable insights into the effectiveness of different architectural approaches for Arabic emotion classification. The consistent superiority of Arabic-specific models reinforces the importance of language-specialized pre-training for morphologically complex languages.

Model Architecture Insights: The performance gap between Arabic-specific and multilingual models suggests that the linguistic complexity of Arabic—including its rich morphology, dialectal variations, and unique emotional expression patterns—requires specialized model architectures trained on Arabic-specific corpora.

Task-Specific Challenges: Our error analysis reveals that the primary difficulties lie in detecting implicit emotional content rather than explicit expressions. This finding has important implications for system deployment, suggesting that additional context or multi-turn analysis might improve performance on ambiguous cases.

Practical Deployment Considerations: MARBERT's balanced performance across all metrics makes it the most suitable choice for production deployment, where consistent reliability is more important than peak performance on specific metrics.

The main challenges identified in our analysis include:

- **Class Imbalance Effects**: The dominance of neutral content in the shared task dataset continues to pose challenges for balanced classification performance.
- **Implicit Expression Detection**: Subtle emotional expressions, particularly sarcasm and implicit hate speech, remain difficult to accurately classify.
- **Dialectal Variation Impact**: Different Arabic dialects introduce additional complexity that current models handle with varying degrees of success.
- **Context Dependency**: Many emotional expressions require a wider conversational or situational context for an accurate interpretation.

We presented a systematic evaluation of transformer-based encoders for Arabic emotion classification in the MAHED shared task. The results show that Arabic-specific models, especially MARBERT, outperform multilingual alternatives in capturing morphological and dialectal nuances. Key challenges remain to detect implicit and sarcastic expressions, handle class imbalance, and address dialectal variation.

Future work will explore lightweight ensembles of Arabic-specific models, advanced training strategies (e.g., curriculum learning), and incorporating broader context or multimodal cues to improve subtle emotion detection.

Limitations: We focused on single-model fine-tuning with limited hyperparameter exploration, which may cap performance.

Acknowledgments

The authors thank the MAHED shared task organizers for providing the dataset(Zaghouani and Biswas, 2025b)(Zaghouani and Biswas, 2025a)(Zaghouani et al., 2024) and the evaluation framework. We also acknowledge the open-source community for the pre-trained models used in this work.

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

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. *Preprint*, arXiv:2402.03216.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Ahmed Murtadha, Alfasly Saghir, Bo Wen, Qasem Jamaal, Ahmed Mohammed, and Yunfeng Liu. 2024. Alclam: Arabic dialectal language model. *Arabic NLP 2024*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

A Detailed Model Specifications

MARBERT Configuration:

```
TrainingArguments(
    output_dir="./checkpoints_marbert",
    eval_strategy="epoch",
    save_strategy="epoch",
    per_device_train_batch_size=64,
    per_device_eval_batch_size=128,
    num_train_epochs=3,
    learning_rate=1e-6,
    warmup_ratio=0.1,
    weight_decay=0.01,
    logging_strategy="epoch",
    save_total_limit=2,
    load_best_model_at_end=True,
    metric_for_best_model="f1_macro",
    greater_is_better=True,
    dataloader_num_workers=2
)
```

Model Architecture Details:

- **MARBERT**: 12 layers, 768 hidden dimensions, 12 attention heads
- **AraBERT-Twitter**: 24 layers, 1024 hidden dimensions, 16 attention heads
- **ALCALM**: 12 layers, 768 hidden dimensions, 12 attention heads
- **XLM-RoBERTa**: 12 layers, 768 hidden dimensions, 12 attention heads
- **LaBSE**: 12 layers, 768 hidden dimensions, 12 attention heads
- **BGE**: Variable architecture depending on specific variant

B Hardware and Runtime Details

All experiments were conducted on GPU-accelerated hardware with the following specifications:

- GPU: NVIDIA Tesla V100 with 32GB memory
- Training time: Approximately 2-4 hours per model for 3 epochs
- Framework: PyTorch 1.12+ with Hugging Face Transformers 4.21+
- Additional libraries: scikit-learn, numpy, pandas

CUET-823 at MAHED 2025 Shared Task: Large Language Model-Based Framework for Emotion, Offensive, and Hate Detection in Arabic

Ratnajit Dhar, Arpita Mallik
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2004008, u2004023}@student.cuet.ac.bd

Abstract

This paper presents our system for Subtask-2: Emotion, Offensive Language, and Hate Detection in the MAHED 2025 Shared Task at ArabicNLP 2025. We address the challenge of multi-label classification in Arabic social media text using a two-stage, prompt-based framework with large language models. In the first stage, our system classifies emotions into 12 distinct categories; in the second stage, it detects offensive messages and, when relevant, further identifies the presence of hate speech. Both stages leverage the Meta-Llama-3.1-8B model, fine-tuned to capture the diverse linguistic and dialectal characteristics of Arabic. Our approach achieved a macro F1-score of 0.518 on the official test set, placing 4th in Subtask 2. The results demonstrate the effectiveness of prompt-based modeling for complex Arabic text classification and contribute a practical, LLM-based solution for emotion and hate speech detection in low-resource scenarios.

1 Introduction

The growing influence of social media platforms has fundamentally transformed how individuals express emotions and communicate across digital spaces, with Arabic-speaking communities having represented one of the fastest-growing user bases globally (Ali and Aleqabie, 2024; Alqahtani and Alothaim, 2022). According to Statista, the internet user population in the United Arab Emirates (UAE) has peaked in 2025 and has increased by almost a thousand users compared to the previous year. This rapid growth has generated vast amounts of user-generated content in diverse Arabic dialects. Consequently, there has been a growing need for robust NLP tools to understand and moderate Arabic content, especially given the mix of emotions and potential for offensive or hateful language. Yet, existing moderation systems have often struggled with Arabics morphological complexity, dialect diversity (Center for Democracy and Technology, 2023), and limited training data and cultural awareness (AL-Sarayreh et al., 2023).

To address these challenges, we have participated in Subtask-2 of the MAHED 2025 Shared Task on Multimodal Detection of Hope and Hate Emotions in Arabic Content (Zaghouani et al., 2025). The purposive focus of this task has been to identify the emotion expressed in Arabic social media text, determine whether the text is offensive, and, if offensive, further assess whether it contains hate content.

To achieve our goal, we have employed a large language model (LLM), specifically a lightweight Meta-Llama-3.1-8B, as the core of our system. This powerful LLM has been fine-tuned using the Unsloth framework, allowing us to efficiently adapt it to the challenging Arabic emotion, offensive language, and hate speech detection tasks. The use of a large language model has enabled us to build an effective system that has achieved competitive performance (macro F1-score: 0.518), ranking 4th among all submissions. The main contributions of this work have been:

- Proposed a two-stage prompt-based framework linking emotion and hate speech detection.
- Applied lightweight LLM fine-tuning for efficient Arabic multi-label classification.
- Showed conditional prompting outperforms flat multi-label methods in low-resource Arabic NLP.

Further implementation details can be accessed via the GitHub repository.[1]

[1] https://github.com/ratnajit-dhar/MAHED

Proceedings of The Third Arabic Natural Language Processing Conference, pages 620–625
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

The detection of hate speech and offensive language in Arabic has remained challenging due to its complex morphology and dialectal variation. The MAHED 2025 Shared Task (Zaghouani et al., 2025) has required systems to analyze short Arabic texts (mostly tweets) and assign multiple labels. The dataset (Zaghouani et al., 2024; Zaghouani and Biswas, 2025b,a) used in this work was introduced at the ArabicNLP 2025 workshop under the MAHED 2025 Shared Task. Examples of input texts and their corresponding output labels are provided in Appendix A.

Early pioneering work has developed foundational methods for Arabic emotion detection utilizing Twitter data in the context of the Egyptian revolution and has found that it was possible to automatically detect emotions from Arabic tweets after appropriate preprocessing (Rabie and Sturm, 2014). Further studies have provided sizable progress through a variety of approaches. A study to collect Arabic dialect datasets by scraping tweets through Olympic hashtags has derived an accuracy of 68.12% with Complement Naive Bayes classifiers (Al-Khatib and El-Beltagy, 2017). Another one has built large-scale COVID-19 datasets with 5.5 million tweets and has achieved an 83% F1-score for emotion classification using LSTM models (Al-Laith and Alenezi, 2021). The advent of transformer-based models has transformed Arabic emotion detection. Research has shown transformer-based models, such as AraBERT, have been superior to traditional machine learning methods (Qaddoumi, 2022). Arabic hate speech and offensive language detection, an equally challenging space, has had progress on large-scale datasets with special tags for vulgarity and hate speech where researchers have achieved F1-scores of 83.2% using state of the art techniques (Mubarak et al., 2020). The integration of emotional knowledge with hate speech detection through multi-task learning frameworks has shown promising results, with studies demonstrating approximately 3% improvement when combining emotional analysis with hate speech detection tasks (Mnassri et al., 2023).

Building on prior work, our system employs a two-stage prompt-based approach using large language models to efficiently address the problem of hierarchical emotion and hate speech detection with limited data and different dialects.

3 System Overview

Our system has been built upon a two-stage, prompt-based classification framework, using Meta-Llama-3.1-8B as the backbone large language model. The main design choice has been to separate emotion classification from detecting hate and offensive speech, providing appropriate prompts and fine-tuning approaches for each stage. An overview of the architecture is illustrated in Figure 1.

In the first stage, we have fine-tuned the model to detect emotion using the prompts that have explicitly highlighted the 12 emotion categories. An example of a prompt that has been used during training and during the inferencing phase follows:

> The following text is an Arabic text. Your task is to classify the emotion expressed in the text into one of the following categories:
> 1. anger, 2. disgust, 3. neutral, 4. love, 5. joy, 6. anticipation, 7. optimism, 8. sadness, 9. confidence, 10. pessimism, 11. surprise, 12. fear
>
> **Text:** {text}
>
> **Response:** {response}

In the second stage, we employed a dedicated prompt asking the model to create a response in two steps, strictly following the task instructions:

> You are given an Arabic text. Your task is to classify whether the text is offensive or not offensive.
> - If the text is offensive, respond with: offensive
> - Then, further classify the text as either:
> - hate (if it expresses hate speech)
> - not_hate (if it does not express hate speech)
> - If the text is not offensive, respond with: not offensive
> - Then, respond with: `not_applicable` for the second classification.
>
> **Text:** {text}
>
> **Response:**
> 1. {offensive}
> 2. {hate}

Based on the hierarchical nature of the task labels, this two-stage design has treated emotion classification as a foundational step, whereas hate speech classification has only been taken into consideration if the text has already been deemed offensive. Such a conditional setup has not only reduced label confusion but also allowed the model to focus on distinct linguistic patterns at each stage. This approach has aligned with prior work showing that hierarchical approaches have had better performance than flat multi-label classification for multi-label tasks as it reduces a complex task into

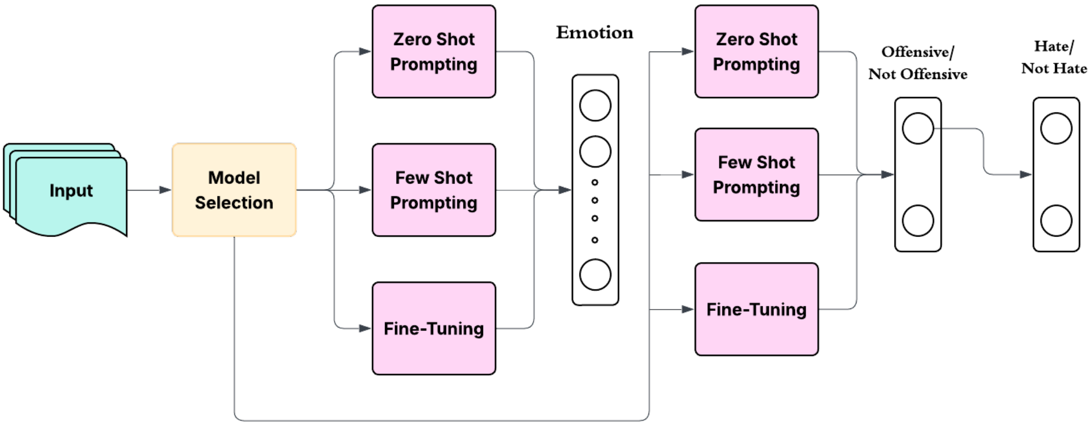


Figure 1: Two-Stage LLM Framework for Arabic Emotion, Offensive, and Hate Speech Detection.

simpler sub-tasks (Galea et al., 2017; Yang et al., 2023). To ensure deterministic outputs during evaluation, we have set the generation temperature to 0.0 for all inference stages.

All fine-tuning and evaluation have used only the official MAHED 2025 dataset; no external data, manual features, or class balancing techniques have been applied.

4 Experimental Setup

4.1 Dataset

We have used the MAHED 2025 dataset from the shared task on Arabic emotion, offensive language, and hate speech classification. Each post is labeled with one of 12 emotions, an offensive label (yes or no), and, if offensive, a hate label (hate or not hate). The training set contained 5,960 posts (1,744 offensive, 303 hate), the validation set 1,277 posts (363 offensive, 68 hate), and the test set consisted of 1,278 unlabeled posts, used solely for final shared task evaluation. The detailed distribution of labels in the training set is shown in Table 1.

Label	**Category**	**Count**
Emotion	Anger	1551
	Disgust	777
	Neutral	661
	Love	593
	Joy	533
	Anticipation	491
	Optimism	419
	Sadness	335
	Confidence	210
	Pessimism	194
	Surprise	143
	Fear	53
	Total	**5960**
Offensive	No	4216
	Yes	1744
Hate (if offensive)	Not Hate	1441
	Hate	303
	Total	**1744**

Table 1: Distribution of emotion, offensive, and hate speech labels in the MAHED 2025 training set.

4.2 Model and Hyperparameters

We have fine-tuned the Meta-Llama-3.1-8B model using 4-bit quantization and LoRA (Hu et al., 2021) with rank 16 for 3 epochs. Emotion and offensive/hate models have been trained for 1000 and 500 steps, respectively, using AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 2e-4, weight decay 0.01, and a linear scheduler with 5 warmup steps. We have employed gradient accumulation and capped input sequences at 2048 tokens. Gradient checkpointing has been enabled to reduce memory usage.

4.3 Libraries and Frameworks

We have used the Unsloth framework (v2024.8) for fine-tuning and the Hugging Face Transformers library (v4.44.0) for model loading/inference. Fine-tuning was configured with TRL, and all models were quantized to 4-bit using BitsAndBytes (v0.43.1) to reduce memory usage.

4.4 Evaluation Metrics

We have used macro-averaged F1-score as the primary evaluation metric. Additionally, we have reported per-class precision and recall to analyze how the model has handled both common and underrepresented emotion and hate categories.

5 Results

In this section, we present the results of our Arabic emotion, offensive language, and hate speech classification task by comparing different prompting strategies as well as models in order to demonstrate their abilities in tackling the problems of this multi-label task.

Our official (fine-tuning LLaMA-3.1-8B) system has had a macro-average F1-score of **0.518** on the test set, placing us in **4th** place out of all of the participating systems.

To gain insight into the impact of different model architectures and prompting strategies, we have compared a number of LLMs using zero-shot, few-shot, and fine-tuned setups. The results, evaluated on the development set, are summarized in Table 2.

In addition to our overall performance, our team (CUET_823) achieved the **highest precision (0.617)**, showing strong effectiveness at minimizing false positives despite slightly lower F1-scores.

Model Name	Prompting Strategy	Macro F1
LLaMA-3.1 8B	Fine-tuning	0.554
	Zero-shot	0.484
	Few-shot	0.477
Mistral 7B v0.3	Zero-shot	0.412
	Few-shot	0.420
	Fine-tuning	0.435
Qwen-2 7B	Zero-shot	0.458
	Few-shot	0.407
	Fine-tuning	0.415
CodeGemma 7B	Zero-shot	0.388
	Few-shot	0.397
	Fine-tuning	0.421
Zephyr 7B	Zero-shot	0.374
	Few-shot	0.386
	Fine-tuning	0.410
Gemma 3 4B	Zero-shot	0.395
	Few-shot	0.402
	Fine-tuning	0.428

Table 2: Macro F1 performance of different models on the validation set.

Fine-tuning has clearly outperformed both zero-shot and few-shot prompting strategies across all models. The LLaMA-3.1-8B model has consistently achieved the highest scores, validating our choice of model and training strategy.

5.1 Error Analysis

While this system has performed very well overall, it has struggled in borderline cases of offensive/the hate speech classification decision. In many cases, the system has flagged text as offensive but has failed to escalate to hate, especially where hatefulness was implied or culturally coded. Below are a few representative errors from the validation set:

Text: شكرا لك وجزاك الله خيرا تحياتي @ZADXII
True: love, no, -
Predicted: neutral, no, -
A positive, polite thank-you message was predicted as neutral rather than 'love' (politeness vs. affection confusion).

Text: العبيد زود انك كريه تسوي ذا الحركات يامريض
True: disgust, yes, hate
Predicted: disgust, yes, not_hate
Racist slur went undetected as hate, showing model's hesitation to escalate from 'offensive' to 'hate' without explicit group targeting.

6 Conclusion

In this work, we have presented a hierarchical two-stage system for Arabic emotion, offensive language, and hate speech detection. The experiments have shown that detecting emotions first and then applying conditional offensive and hate speech classification has been effective due to the strong correlation between these tasks. The hierarchical approach has also been useful in addressing dialectal diversity while being resource-efficient.

Although the system has performed competitively, there have been some limitations. Generalizability has been limited as we have employed only a single dataset and architecture, and static prompting may struggle with evolving language. Not exploring Arabic-specific transformer models may also have limited performance. Future work could explore further architectures, new ensemble methods, dynamic prompting rather than static, wider dialect coverage, and multimodal features for better robustness and contextual understanding.

Acknowledgments

We thank the organizers of the MAHED 2025 Shared Task for providing the dataset and evaluation framework. We are grateful to the anonymous reviewers for their constructive feedback that helped improve this paper.

References

Amr Al-Khatib and Samhaa R El-Beltagy. 2017. Emotional tone detection in arabic tweets. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 105--114. Springer.

Ali Al-Laith and Mamdouh Alenezi. 2021. Monitoring peoples emotions and symptoms from arabic tweets during the covid-19 pandemic. *Information*, 12(2):86.

Sallam AL-Sarayreh, Azza Mohamed, and Khaled Shaalan. 2023. Challenges and solutions for arabic natural language processing in social media. In *International conference on Variability of the Sun and sun-like stars: from asteroseismology to space weather*, pages 293--302. Springer.

Zakaria H Ali and Hiba J Aleqabie. 2024. Emotion detection in arabic text in social media: A brief survey. *Al-Furat Journal of Innovations in Electronics and Computer Engineering*, 3(2):412--421.

Ghadah Alqahtani and Abdulrahman Alothaim. 2022. Emotion analysis of arabic tweets: Language models and available resources. *Frontiers in Artificial Intelligence*, 5:843038.

Center for Democracy and Technology. 2023. Moderating maghrebi arabic content on social media. https://cdt.org/insights/moderating-maghrebi-arabic-content-on-social-media/. Accessed: 2025-08-03.

Dieter Galea, Paolo Inglese, Lidia Cammack, Nicole Strittmatter, Monica Rebec, Reza Mirnezami, Ivan Laponogov, James Kinross, Jeremy Nicholson, Zoltan Takats, and 1 others. 2017. Translational utility of a hierarchical classification strategy in biomolecular data analytics. *Scientific Reports*, 7(1):14981.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2023. Hate speech and offensive language detection using an emotion-aware shared encoder. In *ICC 2023-IEEE International Conference on Communications*, pages 2852--2857. IEEE.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic offensive language on twitter: Analysis and experiments. *arXiv preprint arXiv:2004.02192*.

Abdelrahim Qaddoumi. 2022. Arabic sentiment ensemble nadi shared task 2. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP 2022). Association for Computational Linguistics*.

Omneya Rabie and Christian Sturm. 2014. Feel the heat: Emotion detection in arabic social media content. In *The International Conference on Data Mining, Internet Computing, and Big Data (BigData2014)*, pages 37--49. Kuala Lumpur Citeseer.

Statista. United arab emirates: Number of internet users from 2010 to 2025. https://www.statista.com/statistics/1389944/uae-number-of-internet-users/. Accessed: 2025-08-03.

Youpeng Yang, Qiuhong Zeng, Gaotong Liu, Shiyao Zheng, Tianyang Luo, Yibin Guo, Jia Tang, and Yi Huang. 2023. Hierarchical classification-based pan-cancer methylation analysis to classify primary cancer. *BMC bioinformatics*, 24(1):465.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044--15055.

A Appendix

A.1 Input and Output Examples

The example below shows a sample input and the corresponding output labels from the train dataset:

Example 1

Input: أحد التجار الشباب العمانيين يقول للاسف لما يكون عندهم كاش يروحوا هايبرماركت ولما يريدوا صبر يتسوقوا من عندي!!<LF>متى سندرك أن تسوقنا من تاجر عماني فتح لبيت عماني ودعما لاقتصاد الوطن ، واذا اردتم التأكد فسألوا موظفي البنوك كم من آلاف الريالات يحولها التجار الأجانب إلى الخارج يوميا . https://t.co/tBeNnETQ4z

Output:
Emotion: neutral
Offensive: no
Hate: not applicable

Example 2

Input: RT @tlbakhsh: @AddadRuh مخصصة للاجانب فقط والسعودي نكه!! اشياء عجيبة غريبة مانشوفها غير في السعودية!! المشكلة الاجانب نفسهم في دولهم مايعمل...

Output:
Emotion: anger
Offensive: yes
Hate: yes

Example 3

Input: مسيرات جمعة غضب القدس تتواصل في مختلف مناطق البحرين تنديدًا بخطوة النظام في التطبيع مع العدو الاسرائيلي - ١٨ سبتمبر البحرين#<LF><LF>٢٠٢٠ #التطبيع> $\# < LF >$ $< LF > \#\#\#Bahrainhttps : //t.co/kBhiuqdJfN$

Output:
Emotion: anger
Offensive: yes
Hate: not_hate

AraMinds at MAHED 2025: Leveraging Vision-Language Models and Contrastive Multi-task Learning for Multimodal Hate Speech Detection

Mohamed Zaytoon, Ahmed Salem, Ahmed Sakr, Hossam Elkordi
Department of Computer and Systems Engineering
Alexandria University, Egypt
{mohamed.zaytoon24,es-AhmedMahmod2022,
es-ahmedsakr20,es-hossam.elkordi2018}@alexu.edu.eg

Abstract

Detecting hate speech in social media content is essential to provide a safe space for people to connect. Memes have been used lately to sarcastically express one's opinion, and they can be used to hide harmful intentions and spread hateful speech. In this work, we build our system that detects hateful speech in memes by combining visual and textual features and merging them using different techniques to detect the inherent meaning and overcome the challenge of vast dialectal differences and the variety of topics discussed. To improve our system's robustness, we combine different techniques, such as multi-tasking, contrastive learning, and vision language modeling in a final ensemble model that secured us the third place in the MAHED 2025 shared-task leaderboard with a macro-f1 score of 0.74, showing strong performance on the evaluation set.

1 Introduction

The social media content of the Arabic-speaking world is a complex footprint of social and political expression due to the diverse topics discussed on it, the different points of view introduced, and the different narratives they are presented in. People tend to reflect their hopeful and hateful sentiments on social media platforms, projecting them in different formats of content, such as memes, videos, and textual blog posts (Al-Saqqa et al., 2024; Mulki et al., 2019). The increase of the meme culture over the last few years provided an abundance of multimodal data that introduced nuanced techniques to hide complex and harmful messages through humour and irony (Kiela et al., 2020; Alam et al., 2024a). This necessitates the need for a means of automatic detection of such hateful content to enable safer online platforms for people to express their opinion (Chen and Pan, 2022).

We address the need for robust hateful digital content detection by focusing on the multimodal of Arabic memes (Arya et al., 2024). Such systems must have the capacity to analyze both the visual and textual components of a meme to produce a binary classification of "hate" or "no-hate", thereby mitigating the spread of harmful online content.

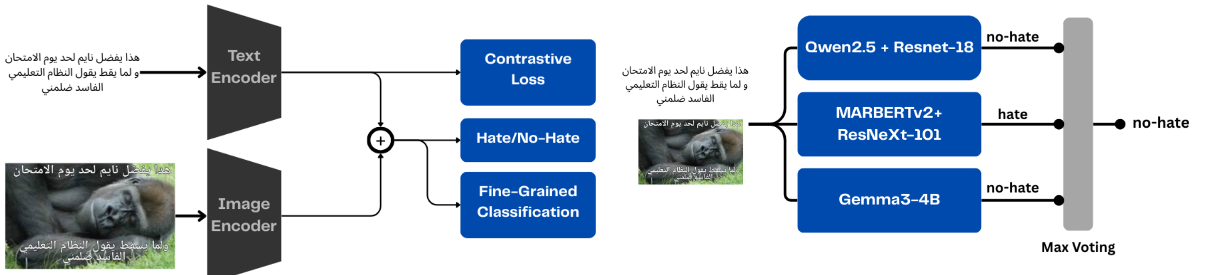
One of the challenges in this task can be the linguistic diversity of Arabic, including vast dialectal variations (Habash, 2010) and cultural expressions with double meanings that can obscure intent (Elkordi et al., 2024). Additional hurdles include the wide range of viral social and political topics, such as politics, religion, and gender, and data-specific issues like the scarcity of clean annotations and a significant imbalance where non-hateful memes are more common (Mulki et al., 2019).

To overcome these obstacles, our system must handle the complexities of the Arabic language by analyzing the interaction between visual and textual features to uncover the actual intent behind sarcastic content. A key requirement is the ability to generalize from limited data across diverse topics and expressions, enabling the model to differentiate between benign cultural commentary and genuinely hateful projections.

Our main contributions can be summarized in the following points:

- Applied multi-tasking technique to benefit from fine-grained classed and contrastive learning to extract meaningful features that can group samples of the same class closer in the embedding space.
- Used a pretrained vision language model in our classification task to benefit from it generalization and multilingual abilities.
- Combined the different techniques we used in a maximum voting ensemble that is robust in multimodal hate speech detection and secured the third place in the shared task leaderboard (Zaghouani et al., 2025).

Proceedings of The Third Arabic Natural Language Processing Conference, pages 626–631
November 8-9, 2025 ©2025 Association for Computational Linguistics



(a) Our dual-encoder training setup in the multi-tasking setup. Both input modalities are encoded separately, then the embeddings are concatenated, and the binary cross-entropy loss is applied on the merged embedding and the text embedding while the contrastive loss is applied only on the textual embedding.

(b) In the testing phase, each model in our system outputs its prediction, then a maximum voting is applied to produce the final prediction. Our ensemble benefit from the different encoding and merging approaches we used with both modalities, including VLMs and encoder/decoder-only models for text embedding.

Figure 1: Illustration of our training in the multi-tasking setup and the maximum voting ensemble in the testing phase

2 Background

Prior research has explored hate speech detection through different classical and deep learning techniques in both unimodal and multimodal settings. The main focus was on the textual content only (Chhabra and Vishwakarma, 2023), relying on classical techniques such as Bag-of-Words (Husain and Uzuner, 2022), TF-IDF (Kumar and Varalakshmi, 2021), Word Embedding e.g. Word2Vec, GloVe, and FastText (Plaza-del Arco et al., 2021), and hybrid methods that combine CNN and GRU or integrate attention mechanisms for improved performance (Zhang et al., 2018).

Then, the focus is switched to deep learning techniques that rely on contextual representation using recurrent networks such as RNNs and LSTMs or BERT-based models (Devlin et al., 2019) that rely on the self-attention technique. More recent work has used both images and text for better contextual representation and accurate results (Kiela et al., 2020). In this setup, multiple techniques have been explored, such as early fusion, late fusion (Lippe et al., 2020), and pre-trained vision language models (Chen and Pan, 2022).

Considering the Arabic language, this is the first use of multimodal memes for hateful speech detection. A previous task explored the use of such a setup for propaganda detection from memes (Hasanain et al., 2024). Participants of this task explored different techniques to integrate visual and textual features to produce the final prediction, such as using multi-agent LLMs to detect the propaganda (Alam et al., 2024a) or using contrastive learning with a multi-objective function (Zaytoon et al., 2024).

3 System Overview

In this section, we present different components of our system. First, we show the backbones and the fusion technique we used in a dual-encoder architecture. Then, we present how we benefited from the instruction capabilities of pretrained vision language models (VLMs) (Bordes et al., 2024). Then, we showcase how we improved the classification performance using a multi-task approach. Finally, we present our system as an ensemble of all the above components.

3.1 Dual-Encoders

In this component, we employed a separate encoder for each modality. For the text modality, we relied on pretrained language models and tested two different approaches. First, we used an encoder-only model, `MARBERTv2` (Abdul-Mageed et al., 2021), which is known for its robustness against dialectal Arabic. We used a version of it that is trained for hate speech detection in the Egyptian dialect (Ahmed et al., 2022). Second, we used a pretrained decoder-only LLM, `Qwen2.5-1.5B` (Team, 2024) to provide a more general representation of the textual input. For the image modality, we used convolutional neural network backbones, specifically `ResNet-101` (He et al., 2016) and `ResNeXt-101` (Xie et al., 2017) models, to capture both global and fine-grained visual features.

After the extraction of both visual and textual features, they are concatenated to form a single multimodal representation. Next, we apply binary cross-entropy on this final representation and the textual embedding, along with a contrastive loss on the textual features only, using in-batch sampling

Split	Number of Samples
Total Training	2,452
Validation	606
Test	500

Table 1: Number of samples in each split of the dataset.

to select positive and negative samples as shown in 1.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BCE}}^{\text{fusion}} + \mathcal{L}_{\text{BCE}}^{\text{text}} + \mathcal{L}_{\text{CL}}^{\text{text}} \quad (1)$$

3.2 Vision Language Models (VLMs)

In this component, we explored the instruction-following capabilities of pretrained VLMs and applied supervised fine-tuning to the recent `Gemma3-4B` model (Team et al., 2025) on our task. The model is provided with an instruction that explains the task and both modalities, text and image. The model is trained in the next token prediction setup, and the cross-entropy loss is applied on the model's output tokens only that include either **"hate"** or **"no-hate"** only.

3.3 Multi-Task

The shared task dataset had another fine-grained classification for each sample, including nine distinct categories. We observed a strong correlation between those categories and the original binary classes. Hence, we decided to benefit from this correlation as an additional supervision by employing a multi-classification objective using our dual-encoder component with two classification heads instead of one.

3.4 Ensemble

Finally, we combined all three components in a maximum voting ensemble as shown in figure 1b, where each of the three models is treated equally and the class with the highest vote is picked as our final classification.

4 Experimental Setup

4.1 Dataset

The dataset is collected from different social media platforms such as Facebook, Instagram, and Pinterest. Then, the dataset went through multiple filteration stages filtered including de-duplication, text extraction, and memes identification. Then, the dataset is randomly sampled from the original 6k samples (Alam et al., 2024b) and gets annotated.

Method	Multi-Tasking	macro-F1
`Qwen2.5-1.5B + ResNet-18`	✗	0.689
`Qwen2.5-1.5B + ResNet-18`	✓	0.702
`Qwen2.5-0.5B + ResNet-18`	✓	0.663
`Qwen2.5-1.5B + ResNeXt-101`	✓	0.699
`MARBERTv2 + ResNet-18`	✓	0.705
`MARBERTv2 + ResNeXt-101`	✓	0.703

Table 2: Macro-F1 results on the validation split using the dual-encoder approach for comparison between single and multi-tasking, as well as the size of both textual and visual encoders and the architecture of the textual encoder.

Table 1 indicates the distribution of the provided dataset. It includes 2,452 samples for training, 606 samples for validation, and 500 samples for final evaluation of the model's performance.

4.2 Training Setup

For the dual-encoders components, we trained the model for 150 epochs with a learning rate of 0.001 and used AdamW optimizer (Loshchilov and Hutter, 2017). We used different batch sizes based on the backbone sizes. We used a batch size of 2 for `Qwen2.5-0.5B` and 16 for `Qwen2.5-1.5B` and `MARBERT`. For the `Gemma3-4B`, we trained the model for 10 epochs with a learning rate of 5e-6 with a cosine scheduler and a batch size of 2. We evaluated our system during training on the validation set using the macro-F1 score, which treats each class equally. All training was done on a single NVIDIA RTX-3090 GPU.

5 Results

In this section, we present a detailed overview of our experiments. All results are reported on the validation set using the macro-F1 score, and finally, we report our test set results on the leaderboard.

5.1 Effect of Multi-Tasking

We conducted our first experiment to assess the effect of the fine-grained categories on the main classification task. We used `Qwen2.5-1.5B` and `ResNet-18` for this experiment. We can see in the first two rows of table 2 that the multi-tasking improved the classification performance by 0.013.

5.2 Size of the Text Encoder

We tested the effect of the text encoder size without changing the image encoder. This experiment was done using the multi-tasking setup. Results

Method	Contrastive Embedding	macro-F1
`Qwen2.5-1.5B + ResNet-18`	text	0.702
`Qwen2.5-1.5B + ResNet-18`	fused	0.666
`Qwen2.5-1.5B + ResNet-18`	text + fused	0.687
`VLM - Gemma3-4B`	-	0.692

Table 3: Comparison between the dual-encoder approach and fine-tuning a pre-trained VLM approach. In the dual-encoder approach, different embeddings were used for the contrastive objective during training.

show that the size increase improved our system performance by 0.039.

5.3 Size of the Image Encoder

Also, we tested the effect of increasing the image encoder size. In this experiment, we compared `ResNet-18` and `ResNeXt-101`. Our results show that the size of the image encoder didn't have a huge improvement on the classification performance.

5.4 Encoder vs. Decoder Models

We tested changing the text encoder architecture and tried using a bi-directional encoder. We used `MARBERTv2` model, and tested it with both `ResNet-18` and `ResNeXt-101`. Using an encoder-only model improved the performance when using different sizes of the image encoder. Also, increasing the size of the image encoder doesn't improve the model's performance.

5.5 Dual-Encoders vs. VLM

Finally, we compared fine-tuning a pre-trained vision language model with the dual-encoder setup with the multi-tasking objective. In the dual-encoder setup, we tested the application of the contrastive objective on different feature vectors: the text embeddings, the image embeddings, and the fused embeddings. We can see that applying the contrastive loss on the text embedding only was the best performing. Also, fine-tuning VLM had very close results and was better than other dual-encoder setups.

5.6 Test Set Results

In this section, we report the results of different systems we submitted and the final maximum voting ensemble we made from them in table 4. We chose the submitted models based on their experimental results shown in tables 2 and 3. Our max-voting ensemble achieved a macro-f1 score of 0.74 and secured our third place in the leaderboard.

Method	macro-F1
`Qwen2.5-1.5B + ResNet-18`	0.71
`MARBERTv2 + ResNeXt-101`	0.72
`VLM - Gemma3-4B`	0.72
Ensemble	**0.74**

Table 4: Macro-F1 scores on the test set submitted on the shared task leaderboard.

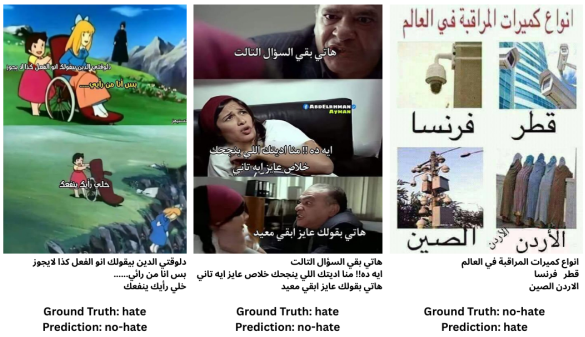


Figure 2: Examples of failure cases from our system.

5.7 Qualitative and Error Analysis

Figure 2 shows cases that our system failed to correctly predict. In the first sample, our system prediction was misguided by the cartoonish scene and failed to identify its hateful stance when discussing religious opinions. In the second image, the obviously sarcastic text over-shadowed the hateful and offensive scene displayed in the image. In the final image, our system identifies the sample as hateful due to its mocking and stereotyping nature.

6 Conclusion

This paper investigated our work in subtask-3 of the MAHED 2025 shared task for hate speech detection in memes. We used contrastive learning and multi-tasking techniques in our dual-enconder component with late embedding fusion. We also fine-tuned a vision language model to benefit from its instruction-following, multi-lingual, and generalization capabilities in the classification task that covers multiple Arabic dialects and different topics. Lastly, we combine different models we built in a robust maximum voting ensemble that secured us the third place in the competition leaderboard with macro-f1 score of 0.74.

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*. Association for Computational Linguistics.

Ibrahim Ahmed, Mostafa Abbas, Rany Hatem, Andrew Ihab, and Mohamed Waleed Fahkr. 2022. Fine-tuning arabic pre-trained transformer models for egyptian-arabic dialect offensive language and hate speech detection and classification. In *2022 20th International Conference on Language Engineering (ESOLEC)*.

Samar Al-Saqqa, Arafat Awajan, and Bassam Hammo. 2024. A survey of hate speech detection for arabic social media: Methods and datasets. *Procedia Computer Science*. 15th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 14th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare EUSPN/ICTH 2024.

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmed, Md Arid Hasan, and Maram Hasanain. 2024b. Armeme: Propagandistic content in arabic memes. *arXiv preprint arXiv:2406.03916*.

Greeshma Arya, Mohammad Kamrul Hasan, Ashish Bagwari, Nurhizam Safie, Shayla Islam, Fatima Rayan Awad Ahmed, Aaishani De, Muhammad Attique Khan, and Taher M Ghazal. 2024. Multimodal hate speech detection in memes using contrastive language-image pre-training. *IEEe Access*.

Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C. Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, Mark Ibrahim, Melissa Hall, Yunyang Xiong, Jonathan Lebensold, Candace Ross, Srihari Jayakumar, Chuan Guo, Diane Bouchacourt, Haider Al-Tahan, and 22 others. 2024. An introduction to vision-language modeling. *ArXiv*.

Yuyang Chen and Feng Pan. 2022. Multimodal detection of hateful memes by applying a vision-language pre-training model. *Plos one*.

Anusha Chhabra and Dinesh Kumar Vishwakarma. 2023. A literature survey on multimodal and multilingual automatic hate speech identification. *Multimedia Systems*.

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*.

Hossam Elkordi, Ahmed Sakr, Marwan Torki, and Nagwa El-Makky. 2024. AlexuNLP24 at AraFinNLP2024: Multi-dialect Arabic intent detection with contrastive learning in banking domain. In *Proceedings of the Second Arabic Natural Language Processing Conference*, Bangkok, Thailand. Association for Computational Linguistics.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Maram Hasanain, Md Arid Hasan, Fatema Ahmed, Reem Suwaileh, Md Rafiul Biswas, Wajdi Zaghouani, and Firoj Alam. 2024. Araieval shared task: propagandistic techniques detection in unimodal and multimodal arabic content. *arXiv preprint arXiv:2407.04247*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Fatemah Husain and Ozlem Uzuner. 2022. Investigating the effect of preprocessing arabic text on offensive language and hate speech detection. *Transactions on Asian and Low-Resource Language Information Processing*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*.

PM Ashok Kumar and K Varalakshmi. 2021. Hate speech detection using text and image tweets based on bi-directional long short-term memory. In *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*. IEEE.

Phillip Lippe, Nithin Holla, Shantanu Chandra, Santhosh Rajamanickam, Georgios Antoniou, Ekaterina Shutova, and Helen Yannakoudakis. 2020. A multimodal framework for the detection of hateful memes. *arXiv preprint arXiv:2012.12871*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Hala Mulki, Hatem Haddad, Chedi Bechikh Ali, and Halima Alshabani. 2019. L-hsab: A levantine twitter dataset for hate speech and abusive language. In *Proceedings of the third workshop on abusive language online*.

Flor Miriam Plaza-del Arco, M Dolores Molina-González, L Alfonso Urena-López, and M Teresa Martín-Valdivia. 2021. Comparing pre-trained language models for spanish hate speech detection. *Expert Systems with Applications*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgio Mikros, Abul Hasnat, and Firoj Alam. 2025. Overview of mahed shared task: Multimodal detection of hope and hate emotions in arabic content. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*. Association for Computational Linguistics.

Mohamed Zaytoon, Nagwa M El-Makky, and Marwan Torki. 2024. Alexunlp-mz at araieval shared task: contrastive learning, llm features extraction and multi-objective optimization for arabic multi-modal meme propaganda detection. In *Proceedings of The Second Arabic Natural Language Processing Conference*.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*. Springer.

DesCartes-HOPE at MAHED Shared task 2025: Integrating Pragmatic Features for Arabic Hope and Hate Speech Detection

Leila Moudjari[1], Mélissa Hacene Cherkaski[1], and Farah Benamara[1,2]
[1] IRIT, Université de Toulouse, ANITI, Toulouse, France
[2] IPAL, CNRS-NUS-A*STAR, Singapore
firstname.lastname@irit.fr

Abstract

This work presents our system for MAHED 2025 Task 1, which focuses on classifying Arabic text into Hope Speech, Hate Speech, or Not Applicable. Our approach combines dialect-aware contextual embeddings with pragmatic features—including speech acts, irony detection, and emotion cues—to capture the nuanced ways in which hope and hate are expressed across diverse Arabic varieties. We also employ targeted data augmentation to improve robustness in underrepresented categories. Experimental results show that incorporating speech act and emotion information significantly enhances detection performance. This approach allowed us to secure the fifth place in the official ranking[1] out of 60 participants, 25 of whom appeared on the final leaderboard, with a macro-F1 score of 0.7010. Our results are promising and mark a first step towards speech-act-aware hope/hate detection for Arabic social media.

1 Introduction

Online hate speech poses serious challenges globally, eroding social cohesion and enabling marginalization. In the Arabic-speaking world, these challenges are exacerbated by the rich tapestry of dialects—such as Egyptian, Levantine, Gulf, Maghrebi—and the frequent use of figurative language, such as irony, that makes automated detection especially complex.

While Arabic hate speech detection has received considerable attention—with resources such as the ADHAR multi-dialect corpus providing richly annotated datasets across both Modern Standard Arabic and major regional dialects, facilitating high-performance classification systems (Charfi et al., 2024)—research on Arabic hope speech detection remains limited compared to other languages. Prior research on hope speech has explored a range of perspectives, from peace-oriented discourse (Palakodety et al., 2019) to multilingual detection for promoting inclusion (Chakravarthi, 2020). Other works have examined expressions of regret and past-oriented hope (Balouchzahi et al., 2023), and the expression of wish in products reviews and political discussions (Goldberg et al., 2009). Building on these prior works, the *CDB model* (Da Silva et al., 2025) introduces a more fine-grained and linguistically grounded classification of hope through counterfactual, desire and belief.

More recently, the EmoHopeSpeech dataset was introduced, a bilingual resource annotated for both emotions and hope speech in English and Arabic, offering fine-grained emotional labels and linguistic variety for deeper analysis (Zaghouani and Biswas, 2025). Additionally, the emergence of innovative computational frameworks has further advanced the study of hope speech: for instance, recent methods leverage emotion-aware modeling to better distinguish hope expressions from neutrality or negativity, However, these approaches remain relatively unexplored in the context of Arabic hope/hate speech detection, and have largely focused on emotions alone (Badawi, 2025).

Building on these insights, the MAHED 2025 Shared Sub-Task1 (Zaghouani et al., 2025) presents an opportunity to jointly study hope and hate speech within a unified framework that accounts for Arabic linguistic variation and rich emotional subtleties. In this work, we extend prior emotion-based approaches by newly incorporating additional pragmatic features—most notably speech acts and irony—alongside emotion categories (e.g., anger, sadness, joy, love). Our system combines dialect-aware transformer embeddings with these pragmatic and affective cues, supported by targeted data augmentation to improve robustness for underrepresented dialects and classes.

Our model achieves a strong F1 score of 0.7010,

[1] Team: IRIT_HOPE.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 632–638
November 8-9, 2025 ©2025 Association for Computational Linguistics

	hope	hate	not applicable	Total
Train-set	1,892	1,301	3,697	6,890
Validation-set	409	261	806	1,476
Test-set	422	287	768	1,477

Table 1: Distribution of the MAHED dataset across training, validation, and test splits, showing the balance of classes for Subtask 1 (hope, hate, and not applicable).

Train Dataset	hope	hate	not applicable	Total
MAHED	1,892	1,301	3,697	6,890
MAHED+subtasks2	1,892	1,604	3,697	7,193
MAHED+MLMA	1,892	2,730	3,697	8,319
MAHED+Synthetic	3,226	1,301	3,697	8,224
MAHED+MLMA+Synthetic	3,226	2,730	3,697	9,653

Table 2: Statistics of the original MAHED train dataset and its augmented variants across the *hope*, *hate*, and *not applicable* classes.

securing fifth place in the official MAHED ranking. In the following sections, we detail our methodology, highlight the impact of speech-act and emotion features, and discuss avenues for advancing hope/hate detection in Arabic social media.

2 Task Overview

MAHED 2025 Task 1 focuses on classifying Arabic social media posts into three categories: hope, hate, or not applicable. The input consists of raw Arabic tweets, encompassing both Modern Standard Arabic (MSA) and various dialects. The output is a single categorical label. For instance:

1. "معاً يمكننا بناء مستقبل أفضل لأطفالنا" (Together, we can build a better future for our children.) → hope

2. "كل المهاجرين لصوص ومجرمون يجب طردهم فوراً" (All immigrants are thieves and criminals who should be expelled immediately.) → hate

3. "اليوم هو يوم مشمس وجميل" (Today is a sunny and beautiful day.) → not applicable

Dataset Details. The MAHED 2025 dataset for sub-task 1 comprises 9,843 annotated instances, divided into training, development, and test sets as shown in table 1.

All instances were collected from public social media platforms, anonymized, and annotated by native speakers, ensuring a Cohen's Kappa agreement greater than 0.85.

3 System Description

Our system for the MAHED2025 shared task builds upon our previous work on exploiting language models for Arabic text classification (Moudjari et al., 2021; Moudjari and Benamara, 2025). In this section, we detail the preprocessing pipeline, data augmentation strategies, and feature integration methods used in our approach.

3.1 Data Augmentation

The MAHED train dataset exhibits a notable class imbalance (see Table 2), with both the *hate* and *hope* categories significantly underrepresented. To mitigate this imbalance and improve model robustness, we implemented targeted data augmentation strategies for these classes.

Hate Class Augmentation. We augmented the hate speech data by incorporating additional annotated instances from two sources:

- **MAHED+subtasks2**: 303 hate-labeled examples were extracted from the second sub-task MAHED: Emotion, Offensive Language, and Hate Detection.

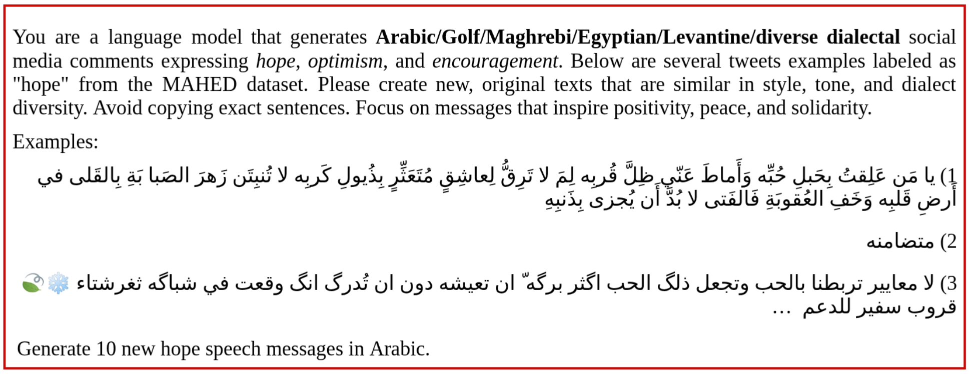
- **MAHED+MLMA**: 1,428 samples annotated with direct offensive and hateful sentiment labels were retrieved from the MLMA dataset (Ousidhoum et al., 2019).

Hope Class Augmentation. Due to the scarcity of publicly available Arabic hope speech datasets, we generated synthetic hope speech data using the ChatGPT-4o language model. By providing in-context examples from the MAHED dataset, we generated 1,334 additional hope-labeled instances designed to preserve domain relevance and linguistic characteristics. The newly augmented dataset is hereafter referred to as **MAHED+Synthetic**.
We instructed the model to generate several hundred Arabic texts covering a wide range of dialects—including Gulf, Egyptian, Maghrebi, and Levantine—and supplemented this with dedicated runs producing several hundred instances for each individual dialect to ensure balanced representation. For each run, we provided dialect-specific examples to guide generation (Figure 1 illustrates the prompts used).

We further combined the newly added inputs from **MAHED+MLMA** and **MAHED+Synthetic** to create **MAHED+MLMA+Synthetic**.

3.2 Enriching Datasets with Pragmatic Features

To provide richer input representations, we automatically augment the original MAHED dataset



You are a language model that generates **Arabic/Golf/Maghrebi/Egyptian/Levantine/diverse dialectal** social media comments expressing *hope, optimism*, and *encouragement*. Below are several tweets examples labeled as "hope" from the MAHED dataset. Please create new, original texts that are similar in style, tone, and dialect diversity. Avoid copying exact sentences. Focus on messages that inspire positivity, peace, and solidarity.

Examples:

1) يا مَن عَلِقتُ بِحَبلِ حُبِّه وَأَماطَ عَنّي ظِلَّ قُربه لِمَ لا تَرِقُّ لِعاشِقٍ مُتَعَثِّرٍ بِذُيولِ كَربِه لا تُنبِتَن زَهرَ الصَبا بَةِ بِالقَلى في
أَرضِ قَلبِه وَخَفِ العُقوبَةِ فَالفَتى لا بُدَّ أَن يُجزى بِذَنبِه

2) متضامنه

3) لا معايير تربطنا بالحب وتجعل ذلگ الحب اگثر برگه" ان تعيشه دون ان تُدرگ انگ وقعت في شباگه ثغرشتاء ❄️🍃
قروب سفير للدعم ...

Generate 10 new hope speech messages in Arabic.

Figure 1: Prompt design for data generation. The prompts guided the model to produce diverse, dialect-rich hope speech texts, ensuring both stylistic variation and balanced representation.

and its augmented variants with emotion, irony, and speech act labels. The emotion and irony annotations follow the configuration described in our previous work (Moudjari and Benamara, 2025) and are detailed in this section, while the speech act annotations follow Benamara et al. (2024). Appendix A provides further details on the datasets used for each annotation type (emotion, irony, and speech acts). Throughout the remainder of this paper, we denote each dataset d (either MAHED or one of its augmented variants) enriched with emotion, irony, and speech act features as d_{emo}, d_{irony}, and d_{SAct}, respectively.

Emotion Detection. We fine-tuned the AceGPT model (Huang et al., 2024) on the **Sem18$_{\text{MSA+Mixed}}$** dataset (Mohammad et al., 2018), which consists of Arabic tweets annotated for eleven emotions (anger, disgust, fear, joy, sadness, etc.). Although the prompting was done in MSA, the dataset itself contains both MSA and various dialectal forms, offering a rich and diverse training resource for emotion classification.

Irony Detection. For irony detection, we fine-tuned the same model on the **IDAT$_{\text{MSA+Mixed}}$** dataset (Ghanem et al., 2019), which consists of Arabic tweets labeled for binary irony classification (ironic vs. non-ironic). Similar to the emotion dataset, it includes a mix of MSA and dialectal varieties, enabling robust evaluation across linguistic registers.

Speech Acts. Following our previous work (Benamara et al., 2024), we employed the `arabertv02-twitter` model (Antoun et al., 2020), fine-tuned on the ArSAS$_{MSA+Mixed}$ dataset (Elmadany et al., 2018), to predict the underlying communicative function of each tweet. The model classifies speech acts into four categories: Subjective, Assertive, Interrogative, and Jussive — corresponding in Arabic to: موضوعي, تأكيدي, استفهامي, and أمري, respectively.

3.3 Model Architecture

Our final architecture builds on `arabertv02-twitter`,[2] a BERT-based model pretrained on a large corpus of Arabic tweets and adapted to the challenges of social media text, including dialectal variation, orthographic inconsistency, and noisy user-generated content. We fine-tuned this model for multi-class classification on the MAHED dataset and its augmented variants (see Table 2). The input text is tokenized and fed into the base model, and class probabilities are produced through a softmax output layer. Training is performed using weighted cross entropy, with early stopping based on the development set F1 score. We train the model for three epochs with a learning rate of $2e-5$, employing the Adam optimizer with an epsilon value of $1e-8$. The batch size is fixed at 16 for training and 128 for validation.

[2] It is worth noting that during the development phase, we submitted several runs using alternative embedding models, including `CAMeL-Lab/bert-base-arabic-camelbert-msa`, `CAMeL-Lab/bert-base-arabic-camelbert-mix`, `SI2M-Lab/DarijaBERT`, and `SI2M-Lab/DarijaBERT-mix`, as well as Arabic-centric large language models such as `FreedomIntelligence/AceGPT-v2-8B` and `FreedomIntelligence/AceGPT-v2-8B-Chat`. Notably, the AceGPT models gave results similar to `bert-base-arabertv02-twitter`. Nevertheless, `bert-base-arabertv02-twitter` proved to be the most effective model in our experiments, and thus we focus our reported results on this model.

3.4 Feature Integration

We explored several strategies for integrating pragmatic cues into the model:

Token-level Augmentation: The most effective approach was to append the predicted emotion, irony, and speech act labels directly to the raw input text prior to tokenization. This method consistently yielded the best performance across our experiments.

Separate Embedding Channels: We also experimented with multi-channel architectures, processing the original text and the additional cues in parallel before merging their representations. However, this approach did not lead to performance gains; in fact, it resulted in a $\sim 2\%$ drop in validation accuracy compared to token-level augmentation.

Normalised Log Feature Scaling: Since the direct insertion of categorical features into text was the most effective, we also experimented with a numeric encoding pipeline for these cues — first normalising label values (z-score), then scaling to $[0, 1]$, and finally applying a log transformation (`normalLog`). While this representation was numerically well-behaved and closer to direct token insertion in terms of accuracy, it did not outperform the plain token-level augmentation approach.

4 Results and Discussion

For all experiments, we used the official MAHED 2025 training set and its augmented version for model fitting and the development set for hyperparameter tuning and model selection. The results reported in Table 3 correspond to performance on the test set as evaluated on the Codabench platform. The final system submitted to the shared task was chosen based on its macro F-score during the development phase, then retrained on the full training data and evaluated by the organizers on the official test set to produce the leaderboard score.

Table 3 presents the experimental results obtained on the MAHED dataset and its augmented variants. Overall, the results show that augmentations incorporating emotion cues (`_emo`) and speech acts (`_SActs`) generally improve performance over the baseline. The best-performing configuration, MAHED+MLMA+Synthetic$_{\text{SAct}}$, reached a macro F-score of 0.7014, outperforming both our MAHED baseline (0.6400) and the official BERT baseline (0.5300). In contrast, adding irony features did not yield consistent gains—either in isolation or in combination with other features—suggesting possible redundancy or the introduction of noise in certain configurations. This outcome can be attributed to the fact that, upon inspection, we found that over 95% of the inputs in the file were labeled as non-ironic.

Test Dataset	F-score
ShardTask baseline	0.5300
Our baseline	0.6400
MAHED$_{\text{emo}}$	0.7000
MAHED$_{\text{irony}}$	0.6900
MAHED$_{\text{SAct}}$	**0.7010***
MAHED$_{\text{emo+irony}}$	0.6800
MAHED$_{\text{SAct+emo+irony}}$	0.6900
MAHED+subtasks2	0.6200
MAHED+MLMA	0.6900
MAHED+Synthetic	0.6800
MAHED+MLMA+Synthetic	0.6900
MAHED+MLMA+Synthetic$_{\text{emo}}$	**0.7007**
MAHED+MLMA+Synthetic$_{\text{irony}}$	0.6934
MAHED+MLMA+Synthetic$_{\text{SAct}}$	**0.7014**

Table 3: Macro F-scores of `bert-base-arabertv02-twitter` fine-tuned on the MAHED dataset and its augmented variants. The score marked with * corresponds to the official leaderboard submission, for which full precision is available. Bolded scores indicate newly obtained runs.

Dataset	hate	hope	not applicable
Our baseline	0.6643	0.5392	0.7081
MAHED$_{\text{emo}}$	0.703	0.6611	0.7237
MAHED$_{\text{SAct}}$	0.7038	0.669	0.7297
MAHED+MLMA+Synthetic$_{\text{emo}}$	0.7078	0.6757	0.7187
MAHED+MLMA+Synthetic$_{\text{SAct}}$	0.7094	0.6635	0.7314

Table 4: Class-wise macro F-scores for the baseline and top-performing augmented configurations on the official MAHED 2025 test set. Scores are reported for each class (*hate*, *hope*, and *not applicable*) along with the overall macro F-score.

Table 4 presents class-wise macro F-scores for the best-performing configurations, alongside our baseline system. In addition to the overall macro F-score, we report separate scores for the *hate*, *hope*, and *not applicable* classes. This breakdown allows us to assess whether specific augmentations, such as emotion or speech act features, offer balanced improvements across all categories or disproportionately benefit particular classes. Emotion features seem especially beneficial for improving the hope class, while speech acts give more balanced improvements across classes, particularly boosting hate and not applicable. To assess whether the observed performance differences between models are statistically meaningful, we conducted McNemar's tests on paired classification outputs. Results revealed no significant differences among most top configurations, except for MAHED$_{\text{SAct}}$ over MAHED+MLMA+Synthetic$_{\text{SAct}}$ ($p = 0.0322$ using McNemar's test), see Appendix Section B for more detailed on these tests.

These findings underscore the importance of pragmatic and affective cues—particularly emotion and speech act information—in detecting hope and hate speech in Arabic social media.

5 Conclusion

We presented an approach to hope and hate speech detection for Arabic social media, leveraging dialect-aware contextual embeddings, pragmatic features (emotion, irony and speech act), and targeted data augmentation. Our results show that dialect sensitivity and augmentation substantially improve performance across Arabic varieties, and that incorporating affective and pragmatic cues—especially speech acts—yields further gains. These findings underscore the importance of modeling both linguistic diversity and communicative intent in fine-grained content moderation. Future work will explore contrastive learning to better disentangle hope and hate in the embedding space, as well as cross-task transfer from sentiment and stance datasets to enrich affective representations and enhance generalization.

Acknowledgments

This work has been supported by DesCartes: the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program. It has also benefited from the AI Interdisciplinary Institute ANITI under the Grant agreement n°ANR-23-IACL-0002.

Limitations

While our system demonstrates improved performance in detecting hope and hate speech across Arabic dialects, several limitations remain. First, the reliance on synthetic data—particularly for the under-represented hope class—introduces a risk of distributional mismatch between generated and naturally occurring texts. Second, our augmentation process covered only four major dialect families; smaller regional varieties remain underexplored. Third, pragmatic features such as irony and speech acts were derived from automatically predicted labels, which may propagate upstream errors into the final classification. Finally, our experiments were limited to the MAHED dataset, and generalizability to other genres (e.g., spoken discourse, formal writing) remains to be validated. Future work will address these issues by expanding dialectal coverage, improving the robustness of feature extraction, and testing cross-domain applicability.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Soran Badawi. 2025. Hopedetect: a multicomponent deep learning framework for hope detection in kurdish language. *The Computer Journal*.

Fazlourrahman Balouchzahi, Sabur Butt, Grigori Sidorov, and Alexander Gelbukh. 2023. Reddit: Regret detection and domain identification from text. *Expert Systems with Applications*, 225:120099.

Farah Benamara, Alda Mari, Romain Meunier, Véronique Moriceau, Leila Moudjari, and Valentin Tinarrage. 2024. Digging communicative intentions: The case of crises events. *Dialogue & Discourse*, 15(1):1–44.

Bharathi Raja Chakravarthi. 2020. Hopeedi: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53.

Anis Charfi, Mabrouka Besghaier, Raghda Akasheh, Andria Atalla, and Wajdi Zaghouani. 2024. Hate speech detection with adhar: a multi-dialectal hate speech corpus in arabic. *Frontiers in Artificial Intelligence*, 7:1391472.

Tulio Ferreira Leite Da Silva, Gonzalo Freijedo Aduna, Farah Benamara, Alda Mari, Zongmin Li, Li Yue, and Jian Su. 2025. Cdb: A unified framework for hope speech detection through counterfactual, desire and belief. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4448–4463.

A Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. Arsas: An arabic speech-act and sentiment corpus of tweets. *Osact*, 3:20.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Véronique Moriceau, and Paolo Rosso. 2019. Idat@fire2019: Overview of the track on irony detection in arabic tweets. In *Mehta P., Rosso P., Majumder P., Mitra M. (Eds.) Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings. In: CEUR-WS.org, Kolkata, India, December 12-15*, pages 10–13.

Andrew B Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. 2009. May all your wishes come true: A study of wishes and how to recognize them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Song Dingjie, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 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 (Volume 1: Long Papers)*, pages 8139–8163, Mexico City, Mexico. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Leila Moudjari and Farah Benamara. 2025. Are dialects better prompters? a case study on arabic subjective text classification. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17356–17371.

Leila Moudjari, Farah Benamara, and Karima Akli-Astouati. 2021. Multi-level embeddings for processing arabic social media contents. *Computer Speech & Language*, 70:101240.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. In *Proceedings of EMNLP*. Association for Computational Linguistics.

Shriphani Palakodety, Ashiqur R KhudaBukhsh, and Jaime G Carbonell. 2019. Hope speech detection: A computational analysis of the voice of peace. *arXiv preprint arXiv:1909.12940*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

A Datasets

Sem18$_{MSA+Mixed}$ (Mohammad et al., 2018): We use the Emotion Classification (E-C) subset from the SemEval-2018 Task 1 “Affect in Tweets” challenge[3]. This dataset contains tweets collected in 2017 and manually annotated into 11 emotion categories: *Anger, Anticipation, Disgust, Fear, Joy, Love, Optimism, Pessimism, Sadness, Surprise*, and *Trust*.

IDAT$_{MSA+Mixed}$ (Ghanem et al., 2019): This dataset comprises tweets on various political issues and events in the Middle East from 2011 to 2018. The tweets are written in Modern Standard Arabic (MSA) as well as Egyptian, Gulf, Levantine, and Maghrebi dialects, and each tweet is manually labeled as Ironic or Not-Ironic.

ArSAS$_{MSA+Mixed}$ (Arabic Speech-Act and Sentiment Corpus of Tweets): is a manually annotated dataset comprising over 21,000 Arabic tweets drawn from diverse dialects and topics. Each tweet is labeled with one of six speech-act categories—Assertion, Expression, Recommendation, Question, Request, and Miscellaneous—as well as one of four sentiment labels: Positive, Negative, Neutral, or Mixed.

B Statistical Significance Analysis

Comparison	p-value
MAHED$_{\text{emo}}$ vs MAHED$_{\text{SAct}}$	0.3173
MAHED+MLMA+Synthetic$_{\text{emo}}$ vs MAHED+MLMA+Synthetic$_{\text{SAct}}$	0.1416
MAHED$_{\text{emo}}$ vs MAHED+MLMA+Synthetic$_{\text{emo}}$	0.3173
MAHED$_{\text{SAct}}$ vs MAHED+MLMA+Synthetic$_{\text{SAct}}$	**0.0322**

Table 5: McNemar’s test p-values comparing top-performing configurations. Statistically significant results ($p < 0.05$) are in bold.

To assess whether the observed differences were statistically significant, we conducted McNemar’s tests between the top configurations. The comparisons between MAHED$_{\text{emo}}$ and MAHED$_{\text{SAct}}$ ($p = 0.3173$), as well as between their augmented counterparts MAHED+MLMA+Synthetic$_{\text{emo}}$ and

[3] https://huggingface.co/datasets/SemEvalWorkshop/sem_eval_2018_task_1

MAHED+MLMA+Synthetic$_{\text{SAct}}$ ($p = 0.1416$), did not yield significant differences, indicating comparable performance. Similarly, the difference between MAHED$_{\text{emo}}$ and MAHED+MLMA+Synthetic$_{\text{emo}}$ was not significant ($p = 0.3173$). However, the comparison between MAHED$_{\text{SAct}}$ and MAHED+MLMA+Synthetic$_{\text{SAct}}$ showed a statistically significant improvement ($p = 0.0322$), suggesting that dataset augmentation benefits speech act–enriched models more consistently.

ANLP-UniSo at MAHED Shared Task: Detection of Hate and Hope Speech in Arabic Social Media based on XLM-RoBERTa and Logistic Regression

Yasmine El Abed[1], **Mariem Ben Arbia**[1], **Saoussen Ben Chaabene**[1], **Omar Trigui**[1,2]
[1]University of Sousse, Sousse 4000, Tunisia
[2]MIRACL Laboratory, Sfax, Tunisia
yasmine.elabed@isgs.u-sousse.tn, mariem.benarbia@isgs.u-sousse.tn,
saoussenbenchabane@isgs.u-sousse.tn, Omar.Trigui@isgs.u-sousse.tn

Abstract

In this paper, we present our system for Subtask 1 of the MAHED 2025 shared task, which involves classifying Arabic text into three categories: Hate, Hope, and not_applicable. Our methodology integrates XLM-RoBERTa embeddings with supervised ML and deep learning techniques. After applying Arabic-specific preprocessing, we extract contextual embeddings and mitigate class imbalance using SMOTE . We then train LR and LSTM classifiers on the augmented features space, supplemented by a similarity calculation with Zero-Shot for prediction validation. The system was evaluated in two phases: using the initial validation set, and the official updated datasets. Results show competitive performance, particularly in boosting recall for minority classes with a macro score of 0.60.

Keywords: Arabic text classification; Hate speech; Hope speech; XLM-RoBERTa; SMOTE; LR; LSTM.

1 Introduction

Hate speech and hope speech are increasingly important phenomena in online discourse, with significant implications for social harmony, policy-making, and content moderation (Ahmad et al., 2024; Charfi et al., 2024). Detecting such speech in Arabic presents unique challenges due to the language's morphological complexity, dialectal diversity, and scarcity of labeled datasets (Haidar, 2021). These challenges are further compounded by class imbalance, where certain categories are underrepresented, leading to biased models and reduced generalization (Haj Ahmed et al., 2024).

Recent works on hope speech detection have shown that transformer-based models combined with transfer learning can achieve competitive results, but most research has focused on English datasets, leaving Arabic-specific challenges underexplored.

The MAHED 2025 shared task (Zaghouani et al., 2025) provides an opportunity to address this gap by focusing on a three-way classification task for Arabic language texts into hate, hope, and not_applicable categories. This task is critical in combating online hostility while also promoting positive and constructive discourse online. However, existing literature and models often focus exclusively on identifying offensive content, neglecting the importance of detecting encouraging or hopeful messages (Chakravarthi et al., 2022). Moreover, many approaches suffer from language imbalance, inadequate handling of class imbalance, and poor generalization across tasks involving morphologically rich languages such as Arabic (Antoun et al., 2020).

To tackle these challenges, we developed a classification pipeline specifically tailored for Arabic. Our pipeline integrates XLM-RoBERTa embeddings, followed by class balancing (SMOTE), and classification using a logistic regression model. We also explored a comparative approach using a Bidirectional LSTM neural network (Conneau et al., 2020).

Our system was submitted to Subtask 1 of MAHED 2025. On the official leaderboard, our final submission achieved a macro Score of 0.60.

2 Background

The MAHED 2025 shared task (Zaghouani et al., 2025) focuses on the multimodal detection of hope and hate emotions in Arabic online content. In particular, Subtask 1, which we participated in, targets a three-way text classification problem. Given short Arabic text segments collected from social media platforms, the system must predict one of three categories: *Hate*, *Hope*, or *Not_Applicable*.

These texts are written in either Modern Standard Arabic or dialectal variations commonly used on social media. They often include informal spelling, emojis, and punctuation typical of online discourse.

- **Hate**: The text contains hate speech or promotes harmful or hostile rhetoric. Example: أي جاهل غبي متخلف مضحوك عليه يقول الشيعه مسلمين

- **Hope**: The text expresses positive sentiment, optimism, or encourages unity and peaceful resolution. Example: آلاتنسوا اهل حلب من دعواتكم اللهم فرج عنهم وارحمهم وغفر لموتاهم اللهم اميييين

- **Not_Applicable**: The text is neutral or unrelated to the conflict context, and does not express hate or hope. Example: آو لانني احب الاشياء الراقيةْ

Several datasets support this shared task, including *So Hateful!* (Zaghouani et al., 2024), *EmoHopeSpeech* (Zaghouani and Biswas, 2025b), and an *Annotated Corpus of Arabic Tweets for Hate Speech* (Zaghouani and Biswas, 2025a). These resources provide the foundation for the MAHED dataset and highlight the novelty of our approach.

Sentiment and emotion analysis in Arabic has gained increasing attention due to the complexity and richness of the language. A number of shared tasks and benchmarks have emerged to foster research in this area.

Recently, the ArAIEval 2022 shared task (Hasanain et al., 2023) addressed Arabic implicit emotion detection using tweets. The participating systems primarily employed transformer-based models such as AraBERT and multilingual BERT, achieving notable performance. In a related context, Daouadi et al (Daouadi et al., 2024) have shown that applying data augmentation alongside fine-tuning transformer models (e.g., ensemble of pre-trained models) can effectively mitigate class imbalance and significantly improve F1 scores in Arabic hate speech detection tasks.

These studies demonstrate the importance of robust pre-processing, balanced datasets, and fine-tuned multilingual models in Arabic text classification tasks.

3 System Overview

To address the challenge of Arabic text classification into *hate*, *hope*, and *not_applicable* categories, we propose a system that combines a multilingual transformer-based model with machine learning techniques, data balancing, and data augmentation strategies. Our system is composed of six steps as follows:

3.1 Data Preprocessing and Tokenization

Arabic sentences were cleaned by removing diacritics, URLs, emojis, elongations, stop words, and rare tokens. The resulting text was normalized and tokenized at the sentence level using XLMRobertaTokenizer, to meet transformer input specifications.

3.2 Data Augmentation via Back-Translation

To address class imbalance in the hope category, back-translation was applied. Sentences were translated from Arabic to English and then back to Arabic using automated translation APIs, producing semantically equivalent yet lexically diverse samples.

3.3 Feature Representation with XLM-R

XLM-RoBERTa (XLM-R) from Hugging Face Transformers was used as the feature extractor due to its effectiveness in multilingual and low-resource settings. Trained on 100 languages, including Arabic, it captures both syntactic and semantic nuances, making it suitable for Arabic social media text containing dialectal variations.

3.4 Label Encoding

To prepare the text data for machine learning, the categorical labels (hate, hope, not_applicable) were converted into numerical format using integer label encoding. This transformation is essential for compatibility with scikit-learn and deep learning frameworks, which require numerical inputs for both training and evaluation. The mapping preserved the original class distribution while enabling efficient optimization of loss functions (for Logistic Regression and LSTM).

3.5 Data Balancing with SMOTE

To mitigate the class imbalance problem, we employed SMOTE (Synthetic Minority Oversampling Technique) on the training set. SMOTE generates synthetic samples of the minority classes (hope and not_applicable) in the embedding space, thus helping the classifier learn more balanced decision boundaries 1.

3.6 Classification Models

To perform the classification task, we trained and evaluated a diverse set of models. For traditional ML approaches, we employed Logistic Regression. While for neural network architectures, we explored LSTM models

4 Experimental Setup

Table 1 presents the class distribution of the training set, which contains 6,890 Arabic sentences distributed across the three categories.

Table 1: Class distribution in the MAHED 2025 training set

Class	Samples
Not Applicable	3697
Hope	1892
Hate	1301

The dataset was released in two phases. In the First phase, we received a training set and an initial validation set. In the Second phase, the organizers updated the validation set and provided an unseen test set.

4.1 Development Phase

4.1.1 Logistic Regression Model Performance in the Developpement Phase

The LR model demonstrated competitive performance in the ternary classification task, achieving a macro F1-score of 0.57 which calculated as the average of class-wise F1-scores: 0.47, 0.51, 0.72. Class-wise metrics reveal nuanced behavior: the model exhibited strong performance for the majority class "Not_applicable" (precision = 0.64, recall = 0.81, F1 $\bar{0}$.72), suggesting effective handling of prevalent patterns. However, minority classes like "Hate" (precision $\bar{0}$.58, recall $\bar{0}$.40) and "Hope" (precision = 0.63, recall = 0.43) showed lower recall, indicating challenges in capturing these instances (see Appendix, Table 2). The confusion matrix further highlights this imbalance, with notable misclassifications of "Hate" and "Hope" samples as "Not_applicable" (see Appendix, Figure 2).

4.1.2 LSTM Model Performance in the Developpement Phase

The LSTM model achieved a macro F1-score of 0.56, reflecting a trade-off between recall and precision across classes. It showed strong recall for minority classes ("Hate": 0.69, "Hope": 0.70), outperforming Logistic Regression in capturing these instances, but with lower precision (0.47, 0.48), indicating higher false positives. Conversely, the majority class "Not_applicable" had high precision (0.71) but low recall (0.44), suggesting conservative predictions and misclassifications to other classes (see Appendix, Table 3). The confusion matrix confirmed these trends, with notable true positives for minority classes but elevated false positives (see Appendix, Figure 3).

4.2 Test Phase

4.2.1 Logistic Regression Model Performance in the Test Phase

The LR model achieved a macro F1-score of 0.57 during testing, demonstrating varied performance across classes. For minority classes, a trade-off between precision and recall was observed: "Hate" showed high recall (0.68) but moderate precision (0.45), while "Hope" had more balanced metrics (0.53 precision, 0.65 recall). The majority class, "Not_applicable", exhibited strong precision (0.70) but lower recall (0.51), indicating conservative predictions(see Appendix, Table 4). The confusion matrix revealed challenges in distinguishing emotional content ("Hate"/"Hope") from neutral cases, with frequent misclassifications favoring the majority class(see Appendix, Figure 4).

4.2.2 LSTM Model Performance in the Test Phase

The LSTM model achieved a macro F1-score of 0.56, reflecting a trade-off between recall and precision across classes. It showed strong recall for minority classes ("Hate": 0.74, "Hope": 0.70), outperforming Logistic Regression in capturing these instances, but with lower precision (0.43, 0.51 respectively), indicating higher false positives (47 and 78 misclassified as "Not_applicable"). Conversely, the majority class ("Not_applicable") had high precision (0.73) but low recall (0.42), suggesting conservative predictions and frequent misclassifications to other classes (216 as "Hate", 252 as "Hope") (see Appendix, Table 5). The confusion matrix confirmed these trends, with 192 true positives for "Hate" and 288 for "Hope", but elevated false positives that reveal the model's tendency to default to neutral classifications (see Appendix, Figure 5).

5 Results

Our system einvolves two distinct approaches for classifying Arabic texts into three categories: a traditional machine learning model (Logistic Regression) and a deep learning architecture (LSTM). The comparative analysis reveals important insights about their respective strengths and weaknesses. For the official phase developpement, we have obtained the following results:

- Macro F1-score for LR: **0.5658**
- Macro F1-score for LSTM:**0.5561**

For the official phase test, we have obtained the following results:

- Macro F1-score for LR: **0.5740**
- Macro F1-score for LSTM: **0.5551**

The logistic regression model offers more transparent decision-making processes compared to the black-box nature of LSTM (see Appendix, Figure 1).

5.1 Error Analysis

The logistic regression classification model achieved a score of 0.6, meaning that 40% of the data was misclassified. To improve performance, it would be beneficial to explore other models like few-shot or one-shot learning, which can better understand the meaning of words and perform classification with minimal training data.
Although our system achieved competitive results on the MAHED 2025 shared task, several systematic misclassifications were observed. A recurrent error pattern was the confusion between *Hate* and *Hope*. For instance, the sentence أْلله يلعن حيطتكم الغبيه صغار الشرقيه وستبغون صغارْ was annotated as *Hate*, but the model predicted *Hope*. This indicates that the presence of certain positive lexical cues can mislead the classifier, even when the overall semantic orientation is hostile. Similarly, the text آحينما انظر الى مثل هؤلاء السعداء في الامم ..آ was misclassified as *Hope* although it was labeled as *Hate*, reflecting the difficulty of capturing sarcasm and figurative expressions. Another frequent source of error was the misclassification of hateful or politically charged content as *Not_Applicable*. Conversely, the system sometimes failed to recognize hopeful messages. For example, آما جمعه الاسلام لن تفرقه السياسه الجزائر المغربْ, which conveys unity and optimism, was annotated as *Hope* but predicted as *Hate*.

5.2 Discussion

Our system was built by combining contextual word embeddings from XLM-RoBERTa with a logistic regression classifier. A comparison with an LSTM classifier did not reveal a substantial improvement, and logistic regression proved to be more stable and consistent across evaluation settings. These results suggest that while multilingual transformers such as XLM-RoBERTa can provide a strong baseline for Arabic text classification.

6 Conclusion

In this work, we have described our participation in MAHED 2025 sub-task 1. We have developed a system which classifies sentences extracted from Arabic social media in three categories : Hate, Hope and not_applicable. Our system is based on a combination of XLM-RoBERTa embeddings with Logistic Regression and LSTM classifiers, augmented by SMOTE for class imbalance and back-translation. For future work, we plan to experiment with Arabic-specific pretrained models such as MARBERT, as well as few-shot and one-shot learning methods to better capture semantic nuances.

References

Ahmad, M., Usman, S., Farid, H., Ameer, I., Muzammil, M., Hamza, A., Sidorov, G., and Batyrshin, I. (2024). Hope speech detection using social media discourse (posi-vox-2024): A transfer learning approach. *Journal of Language and Education*, 10(4):31–43.

Antoun, W., Baly, F., and Hajj, H. (2020). AraBERT: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Chakravarthi, B. R., Muralidaran, V., Hande, A., Philip, J., McCrae, J. P., Buitelaar, P., Ponnusamy, R., Suryawanshi, S., Sherly, E., and Bandyopadhyay, S. (2022). Hope speech detection for equality, diversity and inclusion. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion (LT-EDI 2022)*, pages 378–388, Dublin, Ireland. Association for Computational Linguistics.

Charfi, A., Besghaier, M., Akasheh, R., Atalla, A., and Zaghouani, W. (2024). Hate speech detection with adhar: a multi-dialectal hate speech corpus in arabic. *Frontiers in Artificial Intelligence*, 7.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Daouadi, K. E., Boualleg, Y., and Haouaouchi, K. E. (2024). Ensemble of pre-trained language models and data augmentation for hate speech detection from arabic tweets. *CoRR*, abs/2407.02448.

Haidar, B. (2021). A survey on hate speech detection in arabic social media. *Journal of King Saud University - Computer and Information Sciences*.

Haj Ahmed, A., Yew, R.-J., Minocher, X., and Venkatasubramanian, S. (2024). Navigating dialectal bias and ethical complexities in levantine arabic hate speech detection. *arXiv preprint arXiv:2412.10991*.

Hasanain, M., Alam, F., Mubarak, H., Abdaljalil, S., Zaghouani, W., Nakov, P., Da San Martino, G., and Freihat, A. A. (2023). Araieval shared task: Persuasion techniques and disinformation detection in arabic text. *CoRR*, abs/2311.03179.

Zaghouani, W. and Biswas, M. R. (2025a). An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Zaghouani, W. and Biswas, M. R. (2025b). Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Zaghouani, W., Biswas, M. R., Bessghaier, M., Ibrahim, S., Mikros, G., Hasnat, A., and Alam, F. (2025). MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Zaghouani, W., Mubarak, H., and Biswas, M. R. (2024). So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

A Appendix

A.1 Tables

Table 2: Classification report of development phase for Logistic Regression

Class	Precision	Recall	F1-score
Hate	0.58	0.40	0.47
Hope	0.63	0.43	0.51
Not_applicable	0.64	0.81	0.72

Table 3: Classification report of development phase for LSTM

Class	Precision	Recall	F1-score
Hate	0.47	0.69	0.56
Hope	0.48	0.70	0.57
Not_applicable	0.71	0.44	0.54

Table 4: Class-wise performance metrics

Class	Precision	Recall	F1-score
Hate	0.45	0.68	0.54
Hope	0.53	0.65	0.58
Not_applicable	0.70	0.51	0.59

Table 5: Class-wise performance metrics

Class	Precision	Recall	F1-score
Hate	0.43	0.74	0.54
Hope	0.51	0.70	0.59
Not_applicable	0.73	0.42	0.53

A.2 Figures

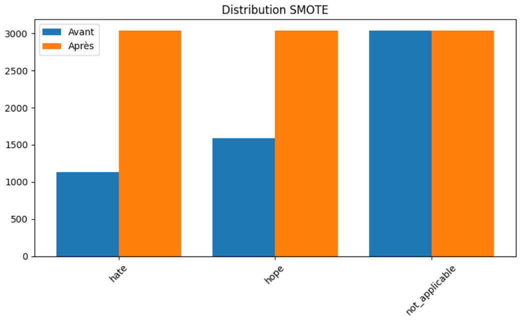


Figure 1: Visualizing Data Balancing with SMOTE

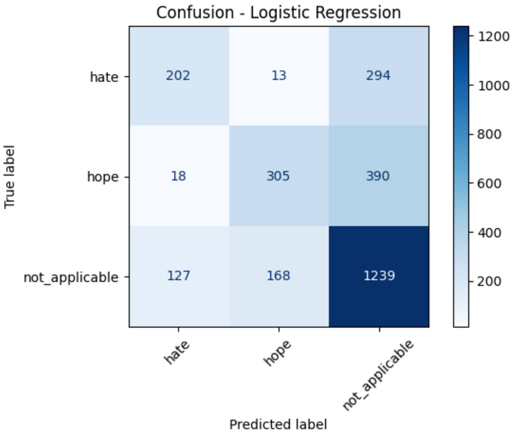


Figure 2: Confusion Matrix of development phase for Logistic Regression

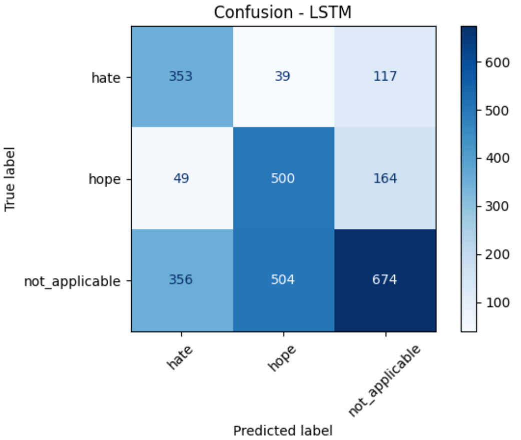


Figure 3: Confusion Matrix of development phase for LSTM

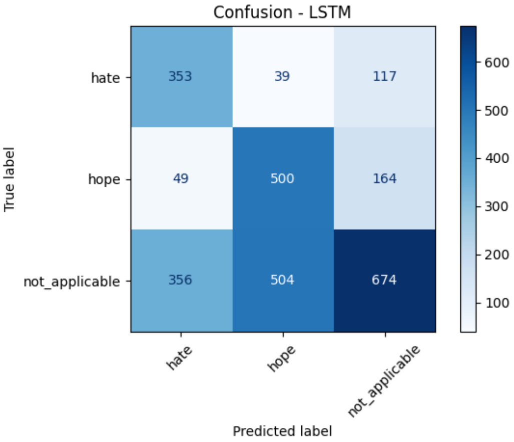


Figure 4: Confusion Matrix of test phase for Logistic Regression

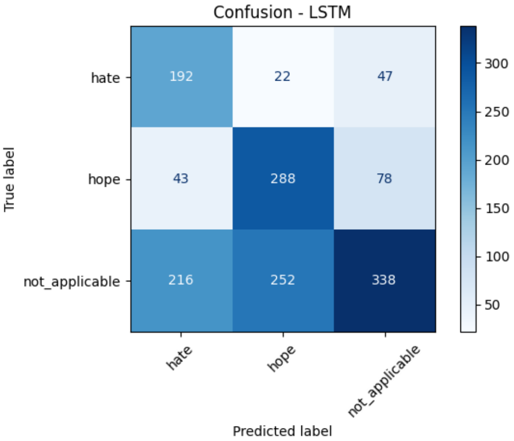


Figure 5: Confusion Matrix of test phase for LSTM

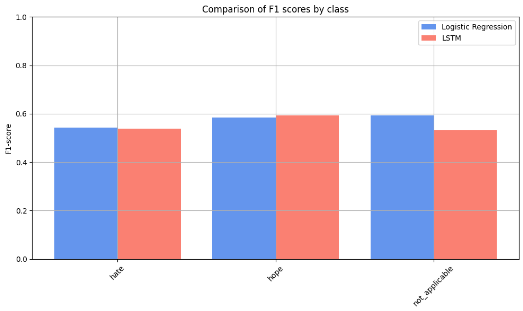


Figure 6: Comparative Analysis of Logistic Regression and LSTM Models

REGLAT at MAHED Shared Task: A Hybrid Ensemble-Based System for Arabic Hate Speech Detection

Nsrin Ashraf[1,2], Mariam Labib[1,3], Tarek Elshishtawy[4], Hamada Nayel[2,5]

[1]Computer Engineering, Elsewedy University of Technology, Cairo, Egypt
[2]Department of Computer Science, Faculty of Computers and Artificial Intelligence,
Benha University, Egypt
[3]Department of Electronics and Communications Engineering, Faculty of Engineering,
Mansoura University, Egypt
[4]Department of Information Systems, Faculty of Computers and Artificial Intelligence,
Benha University, Egypt
[5]Department of Computer Engineering and Information, College of Engineering,
Wadi Ad Dwaser, Prince Sattam Bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia
Correspondence: hamada.ali@fci.bu.edu.eg

Abstract

Hope and hate speech detection in natural language processing addresses the challenge of identifying social media content within the fast-paced environment of online platforms. Hopeful speech that promotes supportive and inclusive language plays a crucial role in counteracting online toxicity, whereas hate speech poses threats and challenges to society. This paper focuses on text-based Arabic hate and hope speech detection, demonstrating the system submitted by the REGLAT team to the MAHED shared task held in conjunction with ArabicNLP 2025. The proposed system employs an ensemble-based model that combines a TF-IDF + Logistic Regression classifier with a fine-tuned AraBERTv2 model as baselines. A majority voting approach is then applied to aggregate the predictions. The proposed model reported an F1 score of 0.58. These promising results are notable given the simplicity of the system's architecture, and they highlight the potential of our approach for improving the performance of this task.

1 Introduction

Hope speech detection in Natural Language Processing (NLP) is a multifaceted research area positioned at the intersection of computational linguistics and artificial intelligence. With the rapid growth of digital communication and social media platforms, the volume and influence of online speech, particularly language that supports mental health and social harmony, have increased significantly (Balouchzahi et al., 2023). Hope speech refers to positive, supportive, and inclusive language that can counteract online toxicity, promote mental health, and offer solidarity to marginalized or vulnerable communities. Detecting and amplifying such speech can play a vital role in mitigating conflict, encouraging resilience during crises, and encouraging inclusive digital spaces (Sharma et al., 2025).

The core objective of hope and hate speech detection is to automatically identify, classify and respond to emotionally charged or socially impactful content within fast-moving streams of user-generated data (Ashraf et al., 2022). Given the scale and speed of online discourse, manual annotation is no longer practical. As a result, modern systems rely on NLP and learning techniques ranging from traditional keywords and lexicon-based methods to more approaches involving machine learning, deep learning, and transformers such as BERT and GPT (Ahmad et al., 2024; ArunaDevi and Bharathi, 2024). These models enable a deeper contextual understanding of language, which is essential for accurately distinguishing between supportive and harmful expressions in diverse linguistic and cultural settings.

The main contribution of this work lies in the development and evaluation of a hybrid approach for Arabic hate speech detection submitted to MAHED shared task (Zaghouani et al., 2025). Arabic is one of the six official languages of the United Nations and has more than 400 million native speakers. Arabic NLP poses unique challenges compared to other languages, due to the complexity of the morphological structure, rich inflection, and diverse dialects (

Proceedings of The Third Arabic Natural Language Processing Conference, pages 645–650
November 8-9, 2025 ©2025 Association for Computational Linguistics

such as Egyptian, Algerian, Tunisian, Gulf Region, Levant, Iraqi etc.) (AbuElAtta et al., 2023; Sobhy et al., 2025). The proposed work combines traditional machine learning techniques with modern transformer-based models. We demonstrate that despite the growing dominance of deep learning, lightweight models such TF-IDF with Logistic Regression remain highly competitive, particularly when complemented by contextual embeddings from models such as AraBERTv2. Furthermore, we propose an ensemble strategy using majority voting to find predictions from both approaches, which yielded the best overall performance in our experiments.

2 Background

Machine learning has played a pivotal role in the advancement of NLP, allowing computers to learn patterns from textual data for tasks such as sentiment analysis, machine translation and text classification (Kamal et al., 2024). Traditional machine learning models, such as Naive Bayes and Support Vector Machines, are heavily based on hand-crafted features and vector representations such as TF-IDF and word embeddings (Khairy et al., 2024).

The rise of deep learning models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNNs) brought significant improvements by automatically learning features from large amount of data. Doghmash and Saad (2025) presented a dual approach to combating hate speech in Arabic social media content. The first part focuses on hate speech detection, where the authors evaluated several deep learning models (RNN, CNN, and CNN-RNN) trained from scratch using AraVec word embeddings (Soliman et al., 2017). Among these, CNN outperforms all other models reported a macro F1-score and accuracy of 0.51 and 0.80 respectively. In contrast, transformer-based models (e.g., QARiB (Abdelali et al., 2021), MARBERT (Abdul-Mageed et al., 2021), AraBERT (Antoun et al., 2020)) significantly outperformed traditional deep learning models. Among these, the QARiB model combined with AraBERT preprocessing achieved the best results, obtaining a 0.92, 0.95 macro F1-score and accuracy respectively.

More recently, transformer-based models such as BERT, MARBERT, and RoBERTa (Liu et al., 2019) have revolutionized NLP by capturing complex contextual relationships through self-attention mechanisms. Abdelsamie et al. (2026) proposed a multi-task learning (MTL) approach using transformer models (AraBERT, MARBERT, MARBERTv2) to improve hate speech detection across Arabic dialects. Each dialect is treated as a separate task to address semantic ambiguity caused by dialectal differences. The AraBERT model learns both shared and dialect-specific features, achieving higher F1-scores than traditional single-task models (up to 0.98 for Egyptian) and 0.85 for MTL combined dialects. It also generalizes well to unseen datasets, proving more effective in detecting hate speech across diverse Arabic dialects.

Daouadi et al. (2024) conducted extensive experiments to optimize hyperparameters and evaluate the effectiveness of transformer-based models for Arabic hate speech detection. Initially, three pre-trained models were fine-tuned using varying parameters. The authors implemented ensemble learning using majority and average voting. These methods further improved performance, with majority voting reaching a weighted F1-score of 0.86. Furthermore, applying data augmentation using external datasets and semi-supervised learning boosted the F1-score to 0.86 and outperformed prior methods across multiple hate speech categories. These models achieve state-of-the-art performance on a wide range of NLP tasks and are now widely adopted in both academic and industrial applications.

3 System Overview

In this study, We experimented with a range of machine learning and deep learning models for text classification. Traditional approaches such as Support Vector Machines (SVMs) have proven effective in handling high-dimensional data, making them well-suited for text classification tasks. Similarly, Logistic Regression (LR) offers a simple yet interpretable linear model that estimates class membership probabilities. Moving beyond these classical methods, we explored Deep Neural Networks (DNNs), which employ multi-layer architectures with ReLU activation functions and dropout regularization to mitigate overfitting. Finally, we fine-tuned transformer-based models, including AraBERTv2 (Antoun et al., 2020) and

CAMeL-BERT (Inoue et al., 2021), both pretrained on large-scale Arabic corpora and shown to be highly effective for Arabic NLP.

3.1 Dataset

The MAHED shared task focusing on the detection of hate speech, hope speech, and emotional expression in Arabic content through three sub-tasks, our team participated in subtask 1 (Zaghouani and Biswas, 2025; Zaghouani et al., 2024). The proposed dataset focuses on classifying Text-based Hate and Hope Speech in Arabic dialects and Modern Standard Arabic (MSA). It consists of manually annotated data collected from social media posts. Each text instance in the dataset is labeled for **`Hate`**, **`Hope`** and **`Not Applicable`**. A general statistics of the class distribution over the dataset is shown in Figure 1. Data were split into training (70%), validation (15%), and testing (15%) sets by the shared task organizers, ensuring consistency and fairness across all participating systems.

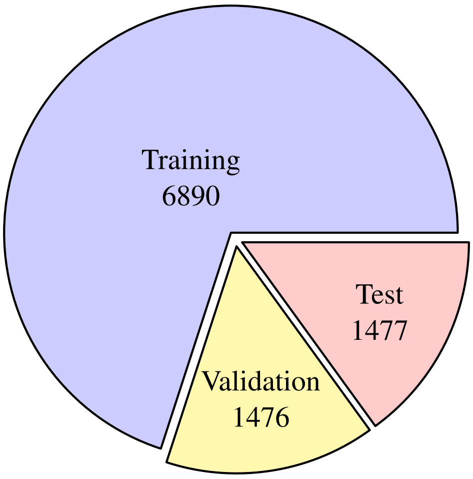


Figure 1: Dataset statistics across training, validation, and test splits.

4 Experimental Setup

In this section, experimental setup, model configurations, dataset preprocessing, evaluation metrics and a detailed analysis of the results have been presented. We have conducted a series of experiments to assess the effectiveness of various pre-trained transformer models and machine learning baselines for Arabic hate speech detection. The experiments are organized into preprocessing, future extraction, hyper-parameter tuning, and evaluation of individual and ensemble models.

4.1 Dataset Preprocessing

A text cleaning strategy using regular expression, implemented using the **`Regex`**, and **`NLTK`** packages. Text cleaning was applied according to the following steps:-

- Remove URLs, mentions, whitespace, punctuation, symbols and emojis
- Normalize Arabic letters
- Remove English letters and numbers

4.2 Model Implementation

This study employed a variety of learning techniques to compare their performance on the same task and determine the impact of each technique on the classification results. These models were selected on the basis of their diversity in mathematical structure and complexity. These models include traditional machine learning, deep learning, and transformer-based models.

For machine learning technique, SVM and LR have been implemented using TF-IDF as a text representation technique over *unigrams*, *bigrams* and *trigrams*. SVM is particularly effective for handling high-dimensional textual data, while LR was used to compute the class probabilities, offering a simple yet robust baseline for text classification tasks. To further enhance the performance of these ML models, a fine-tuning technique was applied using a range of transformer-based models, including `AraBERTv2` and `CAMeLBERT-finetuned-2e-5`.

Deep learning model was applied using multiple hidden layers, incorporating the ReLU activation function and dropout regularization to reduce overfitting. In addition, an ensemble model combining Bi-LSTM and CNN architectures was examined to capture both sequential and local features of the text. This ensemble was tested with two types of word embeddings: a specifically designed for Arabic `AraVec` ($dim = 300$) and a widely used general-purpose embedding `GloVe` ($dim = 200$). These configurations aimed to improve the model's ability to understand semantic and syntactic nuances in Arabic text.

Transformer-based models were applied using pre-trained Arabic language models `AraBERTv2` and `CAMelBERT`. Both models were fine-tuned for sequence classification with three output labels, utilizing the HuggingFace `AutoTokenizer` for tokenization and `AutoModelForSequenceClassification` for model initialization. The training was performed on GPU with parameter alignment through the ignore mismatched sizes option to ensure compatibility. These configurations leveraged the rich contextual representations of Arabic text captured by the transformer models, aiming to enhance classification performance across the **`Hate`**, **`Hope`** and **`Not Applicable`** categories.

Among all the models evaluated, the hybrid technique combining TF-IDF-based Logistic Regression (LR) and transformer-based AraBERT classification, followed by majority voting, achieved the best overall performance. This approach effectively leverages the strengths of both traditional machine learning and deep contextual representations. The LR-based model captured key lexical features, especially effective in high-dimensional sparse text data, while the AraBERT classifier provided deep semantic understanding through pre-trained language representations. To ensure optimal performance, we experimented with different hyperparameter configurations for both models and selected the best-performing settings based on validation results Table 1. In addition, we evaluated several Arabic transformer models and identified AraBERT as the most effective.

Component	Hyperparameters

TF-IDF Vectorizer	max_df = 0.9
	min_df = 5
	max_features = 50000
	ngram_range = (1,3)
	sublinear_tf = True
	norm = ’l2’
	lowercase = True
	stop_words = None

Logistic Regression (LR)	class_weight = balanced
	max_iter = 1000

AraBERT (Transformer)	num_labels = 3 Optimizer: AdamW Learning rate = $2e^{-5}$ Batch size = 16 Epochs = 3 Tokenizer: AraBERT pretrained vocabulary

Table 1: Hyperparameters used in the ensemble model

By applying majority voting between the optimized LR and AraBERT predictions, the system achieved superior classification accuracy and robustness. This ensemble not only mitigated the weaknesses of individual models, but also significantly outperformed standalone deep learning and transformer models, making it the most effective method in our study.

The model was experimented with various configurations to determine the optimal settings for our models. Given that our datasets are imbalanced, as the shared task organizers selected macro F1-Score metric to ensure weight balancing for each label. The final parameters and evaluation metrics are summarized in Table 1.

All experiments were conducted on Google Colab using an NVIDIA T4 GPU with 16 GB of VRAM and 25 GB of system RAM. This setup ensured efficient training and fine-tuning of the transformer-based models while maintaining reproducibility of results.

5 Results and Discussions

This section reports and analyzes the results of our experiments across various configurations and model architectures. We evaluated the performance of traditional machine learning, deep learning, and transformer-based models for Arabic hate speech detection. The results include the impact of hyperparameter tuning and the effectiveness of ensemble learning technique. Our results are very competitive compared to the other teams as shown in Table 2. Among traditional models, SVM and LR achieved macro F1-scores of 0.42 and 0.40 respectively. The CNN-BiLSTM deep learning model underperformed, with a low macro F1-score of 0.31, likely due to limited data or lack of contextual embeddings. However, incorporating transformer-based embeddings significantly improved the results. When combined with **`AraBERTv2`** and **`CAMeLBERT`**, both SVM and LR models showed noticeable performance gains. In particular, **`LR-AraBERTv2`** achieved the highest macro F1-score of 0.58, followed by **`LR-CAMeLBERT`** at 0.54. These results highlight the importance of contextualized transformer embeddings, especially when paired with lightweight classifiers such as LR, to enhance classification performance in Arabic hate speech detection.

Model	Word Representation	Macro F1-score
SVM	TF-IDF	0.42
LR	TF-IDF	0.40
CAMeLBERT	Transformer (AutoTokenizer)	0.36
AraBERTv2	Transformer (AutoTokenizer)	0.46
CNN-BiLSTM	GloVe (200)	0.12
CNN-BiLSTM	AraVec (300)	0.31
SVM-AraBERTv2	TF-IDF + Transformer	0.47
SVM-CAMeLBERT	TF-IDF + Transformer	0.44
LR-AraBERTv2	TF-IDF + Transformer	**0.58**
LR-CAMeLBERT	TF-IDF + Transformer	0.54

Table 2: System performance results on MAHED dataset

6 Limitations

The performance of the models in this study was constrained by several key factors. First, the dataset was imbalanced, which limited the ability of the models to generalize effectively across all classes. Second, the lack of large-scale, domain-specific pre-trained language models for Arabic hate speech reduced the effectiveness of transformer-based approaches, as they struggled to capture the nuanced and context-dependent expressions of hate across dialects. Finally, existing resources for Arabic NLP remain limited compared to high-resource languages, which restricts the range of architectures and embeddings that can be effectively applied.

Future work could address these limitations by developing larger and more balanced datasets, creating domain-specific pre-trained models tailored for hate speech detection, and incorporating dialect-aware modeling. Data augmentation, transfer learning, and multi-task learning also represent promising directions for overcoming data scarcity and improving robustness.

7 Acknowledgment

The ***REGLAT*** team appreciates the efforts that have been offered by Benha University, Elsweedy University of Technology, and Prince Sattam Bin Abdulazizi University.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *Preprint*, arXiv:2102.10684.

Mahmoud Mohamed Abdelsamie, Shahira Shaaban Azab, and Hesham A. Hefny. 2026. The dialects gap: A multi-task learning approach for enhancing hate speech detection in arabic dialects. *Expert Systems with Applications*, 295:128584.

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.

Ahmed H. AbuElAtta, Mahmoud Sobhy, Ahmed A. El-Sawy, and Hamada Nayel. 2023. Arabic regional dialect identification (ardi) using pair of continuous bag-of-words and data augmentation. *International Journal of Advanced Computer Science and Applications*, 14(11).

Muhammad Ahmad, Sardar Usman, Humaira Farid, Iqra Ameer, Muhammad Muzammil, Ameer Hamza, Grigori Sidorov, and Ildar Batyrshin. 2024. Hope speech detection using social media discourse (posi-vox-2024): A transfer learning approach. *Journal of Language and Education*, 10(4):31–43.

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.

S. ArunaDevi and B. Bharathi. 2024. Machine learning based approach for hope speech detection. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), Valladolid, Spain, September 24, 2024*, volume 3756 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Nsrin Ashraf, Mohamed Taha, Ahmed Abd Elfattah, and Hamada Nayel. 2022. NAYEL @LT-EDI-ACL2022: Homophobia/transphobia detection for equality, diversity, and inclusion using SVM. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 287–290, Dublin, Ireland. Association for Computational Linguistics.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Alexander Gelbukh. 2023. Polyhope: Two-level hope speech detection from tweets. *Expert Systems with Applications*, 225:120078.

Kheir Eddine Daouadi, Yaakoub Boualleg, and Kheir Eddine Haouaouchi. 2024. Ensemble of pretrained language models and data augmentation for hate speech detection from arabic tweets. *arXiv preprint arXiv:2407.02448*.

Salam Thabet Doghmash and Motaz Saad. 2025. Arabic hate speech identification and masking in social media using deep learning models and pre-trained models fine-tuning. *Preprint*, arXiv:2507.23661.

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.

Sammer Kamal, Hamada Nayel, and Ahmed Shalaby. 2024. Enhancing hadith classification using statistical and semantic feature fusion and dimension reduction. In *2024 12th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC)*, pages 160–163.

Marwa Khairy, Tarek M Mahmoud, Ahmed Omar, and Tarek Abd El-Hafeez. 2024. Comparative performance of ensemble machine learning for arabic cyberbullying and offensive language detection. *Language Resources and Evaluation*, 58(2):695–712.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Deepawali Sharma, Vedika Gupta, Vivek Kumar Singh, and Bharathi Raja Chakravarthi. 2025. Stop the hate, spread the hope: An ensemble model for hope speech detection in english and dravidian languages. *ACM Transaction Asian Low-Resource Language Information Processing*.

Mahmoud Sobhy, Ahmed H AbuElAtta, Ahmed A El-Sawy, and Hamada Nayel. 2025. Swarm intelligence for handling out-of-vocabulary in Arabic Dialect Identification with different representations. *Neural Computing and Applications*, pages 1–27.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R. El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265. Arabic Computational Linguistics.

Wajdi Zaghouani and Md. Rafiul Biswas. 2025. An annotated corpus of arabic tweets for hate speech analysis. *Preprint*, arXiv:2505.11969.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgio Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

HTU at MAHED Shared Task: Ensemble-Based Classification of Arabic Hate and Hope Speech Using Pre-trained Dialectal Arabic Models

Abdallah Saleh
Al Hussein Technical University
abdullahhsalehds@gmail.com

Mariam Biltawi
Al Hussein Technical University
Mariam.Biltawi@htu.edu.jo

Abstract

Emotional contagion, the phenomenon where emotions spread between individuals, shows the importance of detecting both hope and hate speech in digital communications. This emotional transmission can amplify positive sentiments that foster community resilience or propagate harmful content that divides societies. While hate speech detection in Arabic has been extensively studied, hope speech detection has received comparatively limited attention, creating an imbalance in the understanding of emotional influence online. To address this gap, MAHED 2025 sub-task 1 introduced the task of detecting both hope and hate speech using a substantial dataset designed for developing robust classification models. This paper presents an ensemble approach combining three Transformer-based encoder models with soft voting and weighted loss functions to address class imbalance issues. Those models, ArabicDeBERTa-DA, BERT-DA, and MARBERTV2, have been continually pre-trained on different domains of Arabic, showing the benefits of continual pre-training both on downstream performance and computational efficiency. The proposed ensemble model achieved the highest performance in the competition with an F1 macro score of 72.3% using an ensemble voting of the best-performing variants.

1 Introduction

Arabic NLP researchers have extensively investigated the problem of hate speech in Arabic, particularly through the construction of datasets and the development of detection systems. In contrast, hope speech has received considerably less attention, with only a few datasets available and, consequently, fewer detection systems. sub-task 1 of MAHED 2025 (Zaghouani et al., 2025) seeks to address this gap by introducing a dataset that incorporates both hate speech and hope speech.

The dataset encompasses two varieties of Arabic: Modern Standard Arabic (MSA) and Dialectal Arabic (DA). This diversity introduces significant challenges in developing robust detection systems, as the linguistic variation between these forms of Arabic is substantial. Most pre-trained Arabic language models have been trained almost exclusively on a single domain, which limits their ability to generalize effectively to this dataset. Given the high computational cost of pre-training from scratch, it is often impractical to train a new model entirely for an unseen domain. A practical alternative is continual training, whereby a pre-trained model is further adapted to a new domain through additional pre-training, not only gaining the ability to generalize to a new domain but also retaining performance on the original domain, if done correctly.

Only a limited number of Arabic language models have undergone continual pre-training, with notable exceptions such as MARBERTV2(Abdul-Mageed et al., 2021) and AraBERTv0.2 Twitter(Antoun et al., 2020). MARBERTV2 is an enhanced version of the original MARBERT model, specifically designed to better capture the multilingual and multi-dialectal nature of Arabic text. It was continually pre-trained on diverse Arabic corpora thereby improving its robustness for cross-dialectal tasks. Similarly, AraBERTv0.2 Twitter was continually pre-trained on Twitter data (specifically, approximately 60 million tweets), which enables it to better handle the DA semantic and syntactic language patterns prevalent on social media. This specialized pre-training makes it particularly well-suited for social media text classification tasks, such as the detection of hate and hope speech in informal Arabic discourse.

Ensemble learning in deep learning represents a powerful paradigm that combines predictions from multiple models to achieve superior performance compared to any individual model. This approach leverages the principle of diversity among models, where different models, architectures, training procedures, or data representations can capture

Proceedings of The Third Arabic Natural Language Processing Conference, pages 651–657
November 8-9, 2025 ©2025 Association for Computational Linguistics

complementary/independent patterns and their corresponding errors in the data(Goodfellow et al., 2016). If models are diverse enough, which might translate to their errors being independent, ensemble models will perform significantly better than their members/submodels (Goodfellow et al., 2016). In the context of transformer-based models like BERT variants, ensemble methods can combine models that have been trained on different domains, use different tokenization algorithms or sizes, or have been fine-tuned with a range of hyperparameters. The ensemble typically aggregates predictions through techniques such as majority voting for classification tasks, weighted averaging of probability distributions, or more sophisticated methods like stacking, where a meta-learner is trained to optimally combine the base models' outputs.

The main contribution of this paper is the introduction of two continually pre-trained models, BERT-DA and ArabicDeBERTa-DA, which have been continually pre-trained on DA data. These models, when combined with MARBERTV2 in an ensemble of pre-trained language models, achieved state-of-the-art performance in sub-task 1 of MAHED 2025. The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 describes an overview of the proposed system and methodology, Section 4 presents the experimental setup, Section 5 presents the results, and Section 6 provides a discussion and conclusion.

2 Background

(Ke et al., 2023) have shown that continual pre-training of a general-domain language model to a specific domain increases the performance of the further pre-trained model on the target domain. They have proposed the continual domain-adaptive pre-training, continual DAP for short, methodology to allow for general-domain language models to continue training on domain specific data, ensuring both learning from the new domain, while avoiding catastrophic forgetting of the model's general knowledge.

(Lee et al., 2023) have shown, at least within the domain of computer vision, that continual learning doesn't always bring an increase in performance, especially in strong pre-trained models like CLIP. It was also shown that the algorithm used to training doesn't always have the performance boost when used on continued training phase.

(Ibrahim et al., 2024) have noticed that while adapting language models to new domain knowledge is more data efficient, the process of continued pre-training is challenging. As a sub-optimal continued pre-training can lead to either catastrophic forgetting of previously trained general knowledge of the model or poor performance on the new domain. Hence, different learning rate schedulers have been proposed to allow the model to learn without forgetting its past knowledge. The authors have shown one of the most efficient and simplest way to continually pre-train is to use a simple learning rate scheduler.

Continual pre-training remains a relatively underutilized strategy in Arabic NLP. While models such as AraBERT (Antoun et al., 2020) and MARBERTV2 (Abdul-Mageed et al., 2021) exemplify its application, explicit discussion of continual pre-training is scarce in the broader literature. AraBERT was extended via continual pre-training to adapt to domain-specific nuances such as social media, as in the AraBERTv02-Twitter variant, while MARBERTV2 further refines the original MARBERT by additional pre-training on MSA corpus with longer sequence lengths.

(Sarkar et al., 2022) have also investigated parameter and data efficient continual pre-training approaches for the Arabic language, showing that further adaptation of pre-trained multilingual models (mBERT) with DA can have robust performance. Beyond these instances, continual pre-training in Arabic remains largely unexplored.

(Anezi, 2022) addressed the need for an intelligent system that can detect hate speech in Arabic, as the author highlighted its importance for national security and combating issues like cyberbullying. The author introduced a new dataset of 4,203 Arabic social media comments, which are classified into seven distinct categories: content against religion, racist content, content against gender equality, violent content, insulting/bullying content, normal positive comments, and normal negative comments. This dataset is notably larger and more granular in its classification than most existing Arabic hate speech datasets. The core of the study is a proposed deep recurrent neural network (RNN) model, called DRNN-2. The model's performance was evaluated on three different classification tasks: binary (positive vs. negative), three-class (positive, negative, and hate speech), and the full seven-class classification. The DRNN-2 model achieved a training accuracy of 99.73% for binary classification, 95.38% for the three-class task, and 84.14% for the seven-class task. These results are reported to be higher than those of similar methods in the current literature, demonstrating the model's effectiveness in tackling the complexities of Arabic text and providing a potentially valuable tool for monitoring online content.

(Almaliki et al., 2023) were among the first researchers to pre-train language models and fine-tune them for the task of hate speech detection. They introduced Arabic BERT-Mini Model (ABMM), a smaller BERT variant with a reduced hidden dimension compared to BERT. After pre-training, it was fine-tuned on a newly released dataset of around 9,500 labeled hate speech documents. Despite its modest size estimated to be approximately 11 million parameters, it outperformed much larger models such as AraBERT.

(Gandhi et al., 2024) also delved into the importance of detecting hate speech, while discussing literature about the topic since 2020. Alongside the comprehensive literature review, they introduced a methodology to tackle multi-label and multi-class hate speech in Indonesian. With the LSTM model attaining the highest accuracy compared to Logistic Regression models.

(Chakravarthi, 2022) proposed a different alternative to suppression of hate speech, namely the promotion and assistance of hope speech. In their study, the authors proposed the first multilingual hope speech datasets collected from YouTube along with a novel deep learning architecture to train on this dataset and detect hope speech. The dataset included English, Tamil, and Malayalam text. The multi-annotator annotation process yielded consistent results across annotators, proved by the fairly high inter-annotator agreement of 0.6+ score on Krippendorff's Alpha metric and reaching as high as a near-perfect agreement of 0.85 on labeling Malayalam text. After dataset collection and annotation, they tested it on a suite of machine learning algorithms, ranging from traditional SVM and Logistic Regression models, among others, to their proposed deep learning-based system with a CNN model with T5-sentence embedding and IndicBERT. The proposed model outperformed all other models with an F1 score of 0.75, 0.62, and 0.67 on English, Tamil, and Malayalam, respectively.

Although many hate speech datasets exist, including in Arabic, few researchers have developed a corpus with hope speech, let alone one with Arabic text in it. (Zaghouani and Biswas, 2025b) were among few Arabic NLP researchers to have collected text and created a hope speech dataset. It is a bilingual Arabic-English dataset with around 38,000 data points, collected from social media sites. Not only does it annotate text on emotion labels, but also for intensity, complexity, and cause, categorizing hope speech with both binary and more granular labels. Its reliability can be attributed to its high inter-rater agreement, with a very good - near perfect - agreement, ranging from 0.75 - 0.85 Fleiss' Kappa.

(ArunaDevi and Bharathi, 2024) went beyond just simply creating hope speech datasets, acknowledging the importance of automated and intelligent systems in detection of hope speech, especially in social media platforms due to the "snowball effect" of speech in social media in general Where, according to a Facebook (now Meta) study, emotional states found in comments can directly influence the emotional state of users reading those comments (Kramer et al., 2014). Hence, having systems that can detect hope speech can be of immense usefulness to foster a positive environment where certain types of speech are promoted, further influencing a positive emotional state of their users. The authors proposed different intelligent systems to detect hope speech, ranging from traditional machine models like Multinomial Naive Bayes classifier and Support Vector Machines to the usage of BERT. Unsurprisingly, the BERT model outperformed traditional machine learning models while Multinomial Naive Bayes had a respectable performance.

3 System Overview

The MAHED sub-task 1 (Zaghouani et al., 2024; Zaghouani and Biswas, 2025b,a) dataset comprises MSA and DA instances, split into training, validation, and test sets by the task's organizers. Each text is accompanied by a target label, with possible labels being either "hate", "hope", or "not_applicable". The dataset exhibited somewhat of a class imbalance, with the training dataset having 3,697 text instances labeled as "not_applicable", 1,892 text instances labeled as "hope", and only 1,301 text instances labeled as "hate". Text preprocessing involved the removal of English, emoticons, symbols, and Arabic diacritics. The clean text was then tokenized with a sequence length of 128 tokens, with sentences shorter or longer being padded or truncated, respectively. The tokenized data was stored for later usage.

Pre-trained language models ArabicDeBERTa-DA, BERT-DA, and MARBERTV2 were used. ArabicDeBERTa and BERT-MSA were first pre-trained on a large corpus of approximately 2.2 billion MSA tokens, using Masked Language Modeling (MLM) task, with cross entropy loss as the loss function. The pre-training's aim is for the model to gain general language understanding, which can be leveraged later on downstream tasks, increasing performance. After pre-training exclusively on MSA data, the models have gained knowledge about the syntax, morphology, and semantics of MSA. To facilitate understanding in DA tasks, ArabicDeBERTa-DA and BERT-DA were continually pre-trained on DA text obtained from (Al-Fetyani et al., 2023). While the previous models were first trained on MSA data then DA data, MARBERTV2 was first trained on a large corpus of DA data, then continually pre-trained on MSA data. MARBERTV2 first pre-training phase included around 15.6 billion tokens. Its continual pre-training phase with MARBERTV2 also used MLM as the task type, with cross entropy loss as the loss function.

A custom ensemble model was designed to integrate three transformer-based models: BERT-DA, ArabicDeBERTa-DA, and MARBERTV2, each equipped with a classification head consisting of a linear layer, Tanh activation, and a final linear layer mapping to three output classes. The ensemble combined their outputs through soft voting by averaging logits to produce the final prediction.

4 Experimental Setup

The model was trained for 5 epochs with a batch size of 16, using the AdamW optimizer with a learning rate of 7×10^{-6}. Cross-entropy loss with class weights was applied to address class imbalance. Validation was performed periodically during training.

Performance was evaluated using macro-averaged precision, recall, F1-score, and accuracy on the validation and test sets, demonstrating the effectiveness of the ensemble in multi-class classification.

5 Results

The ensemble model performed at its best within ¾ of the first epoch, achieving a 72.2% F1 score. To achieve the best performance across different ensemble models, different seeds for initialization were set for each ensemble model and then the best performing ensemble models runs were hard voted for submission, achieving 72.3%. The Macro F1 score achieved by this model gained it its first place position in the competition. Each model achieved around 67% F1 score on test dataset, hence their combined performance boosted the score by 5-6%. The addition of a weighted loss function with 1.02 for hate, 0.98 for hope and 1 for not applicable allowed for higher weighting of error for misclassification of the minority class. The different weight initialization allowed for better diversity of heads. The proposed system has close to 0.5 billion parameters. Appendix A includes a confusion matrix of models performance on test set along with error analysis of 3 text instances.

6 Conclusion

This work highlights the importance of detecting both hope and hate speech in Arabic digital communication, as emotional contagion can either foster resilience and harmony or amplify societal divides. To address this challenge, this paper proposed an ensemble model integrating three continually pre-trained transformer-based encoder models: BERT-DA, ArabicDeBERTa-DA, and MARBERTV2. By leveraging continual pre-training on dialectal Arabic and combining models through soft voting with weighted loss functions, the proposed system achieved state-of-the-art results in sub-task 1 of MAHED 2025, with the the ensemble of best-performing models obtained a macro F1-score of 72.3%.

The results demonstrate the effectiveness of ensemble learning in handling linguistic diversity across Arabic varieties along with the performance boost and computational efficiency of continually pre-trained models. Beyond competition performance, the contribution of this work lies in introducing dialect-aware pre-trained models that can be extended to a wide range of downstream tasks. Future research can build on this by exploring larger-scale pre-training and continual pre-training efficacy and transferability between different Arabic dialects along with their downstream performance on tasks such as MAHED 2025 sub-task 1.

Limitations

A key limitation of this study is its proposed model's large size, having a nearly half a billion parameter model makes it unfeasible to run in most edge devices and CPU environments without significant delay in response. This makes the model impractical to run in production and resource-constrained environments.

Acknowledgments

The authors express gratitude to Al Hussein Technical University for its support and for fostering a research environment committed to innovation.

Research supported with Cloud TPUs from Google's TPU Research Cloud (TRC).

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.

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith Abandah, Adham Alsharkawi, and Maha Dawas. 2023. Masc: Massive arabic speech corpus. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 1006–1013. IEEE.

Malik Almaliki, Abdulqader M Almars, Ibrahim Gad, and El-Sayed Atlam. 2023. Abmm: Arabic bert-mini model for hate-speech detection on social media. *Electronics*, 12(4):1048.

Faisal Yousif Al Anezi. 2022. Arabic hate speech detection using deep recurrent neural networks. *Applied Sciences*, 12(12):6010.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

S ArunaDevi and B Bharathi. 2024. Machine learning based approach for hope speech detection. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), CEUR-WS. org*.

Bharathi Raja Chakravarthi. 2022. Hope speech detection in youtube comments. *Social Network Analysis and Mining*, 12(1):75.

Ankita Gandhi, Param Ahir, Kinjal Adhvaryu, Pooja Shah, Ritika Lohiya, Erik Cambria, Soujanya Poria, and Amir Hussain. 2024. Hate speech detection: A comprehensive review of recent works. *Expert Systems*, 41(8):e13562.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*.

Adam DI Kramer, Jamie E Guillory, and Jeffrey T Hancock. 2014. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790.

Kuan-Ying Lee, Yuanyi Zhong, and Yu-Xiong Wang. 2023. Do pre-trained models benefit equally in continual learning? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6485–6493.

Soumajyoti Sarkar, Kaixiang Lin, Sailik Sengupta, Leonard Lausen, Sheng Zha, and Saab Mansour. 2022. Parameter and data efficient continual pre-training for robustness to dialectal variance in arabic. *arXiv preprint arXiv:2211.03966*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

A Results analysis

Figure 1 shows the performance of the best results against the test dataset. Hate speech has been rarely predicted as hope speech, and vice versa.

Table 1 shows examples of text along with their actual and predicted labels, as Error Analysis. Examples of hate speech and overtly vulgar sentiment were avoided. The examples call attention to the ambiguous nature of the text, highlighting the challenging nature of the task along with annotating it.

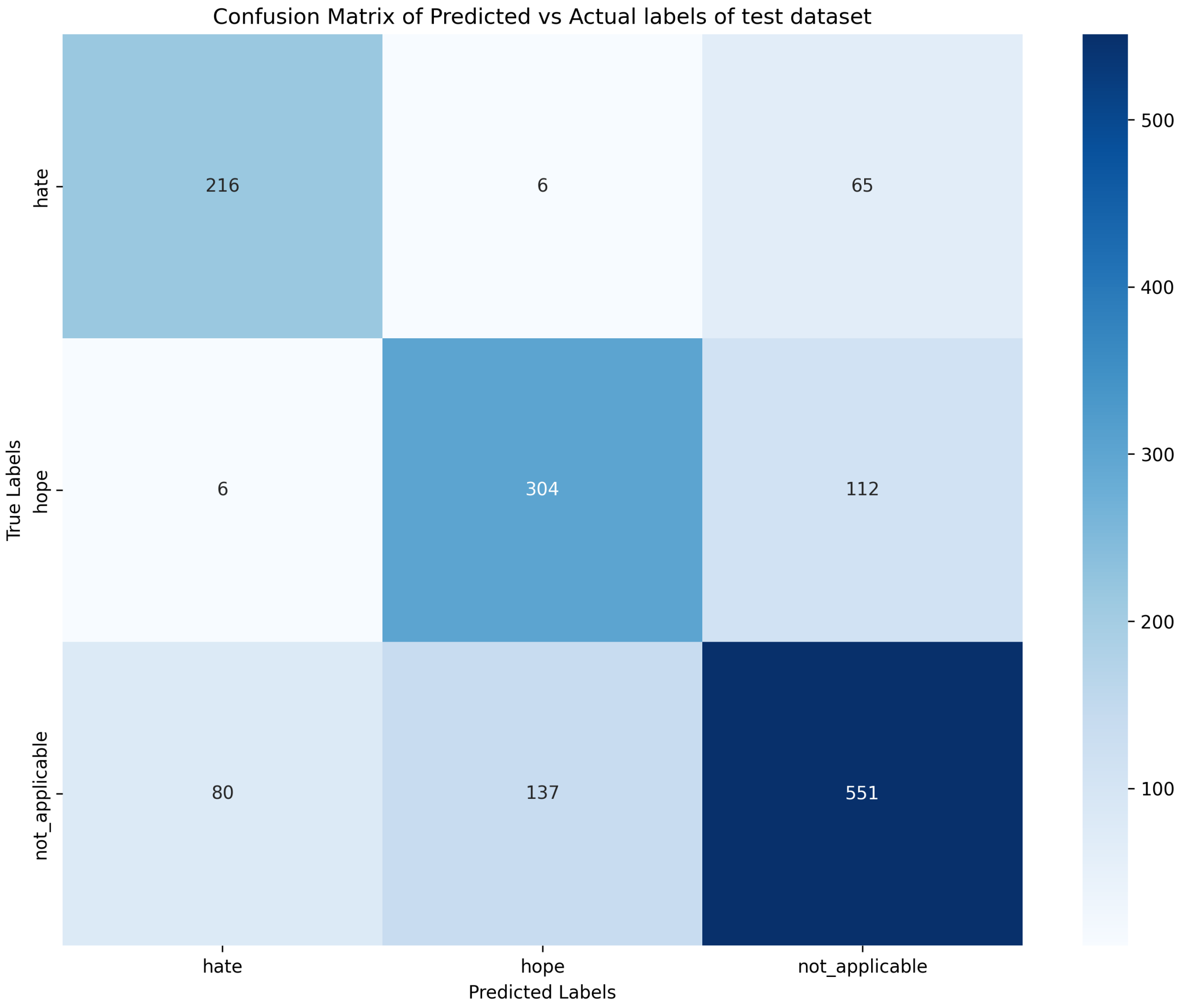


Figure 1: Confusion Matrix displaying performance of best predicted results vs actual labels by class/

Ground Truth	Predicted	Translated Text	Arabic Type
Hope	Not applicable	Oppression Oh God, I seek refuge in You from the oppression of men	DA
Not applicable	Hope	It is not just words of love and flirtation, but rather it is caring and taking care of the one you love and staying with him for a lifetime without changing your feelings towards him, friendship.	MSA
Not applicable	Hope	I put my heart between your hands and swore not to care.	MSA

Table 1: Examples of ground truth vs predictions with Arabic text. The Arabic text was translated using Google Translate.

SmolLab_SEU at MAHED Shared Task: Do Arabic-Native Encoders Surpass Multilingual Models in Detecting the Nuances of Hope, Hate, and Emotion?

Md. Abdur Rahman[1] **Md Sabbir Dewan**[2*] **Md. Tofael Ahmed Bhuiyan**[1*]
Md. Ashiqur Rahman[3]
[1]Department of Computer Science and Engineering, Southeast University, Bangladesh
[2]School of IT, Murdoch University, Australia
[3]School of Computer and Cyber Sciences, Augusta University, Georgia, USA
2021200000025@seu.edu.bd, 35254922@student.murdoch.edu.au,
tofael1104@gmail.com, mdrahman@augusta.edu

Abstract

The dynamic interplay of hope and hate speech on Arabic social media presents a critical challenge for content moderation and digital discourse analysis. This paper presents our systems for the MAHED 2025 shared task on Multimodal Detection of Hope and Hate Emotions in Arabic Content, addressing the two text-based subtasks. Our approach centers on a systematic, empirical comparison of Arabic-native versus large-scale multilingual Transformer encoders to determine the optimal pre-training strategy for this nuanced domain. Comprehensive evaluations demonstrate the clear superiority of Arabic-native models, with our ARBERTv2-based system achieving the highest performance. We secured 11th place in Subtask 1 with a macro F1-score of 0.682 and 5th place in Subtask 2 with a macro F1-score of 0.514. Error analysis reveals persistent challenges in interpreting implicit language and overcoming severe class imbalance, particularly in distinguishing targeted hate from general offensiveness. This work contributes a robust benchmark for this comparison and underscores the importance of language-specific pre-training for nuanced affective computing in Arabic.

1 Introduction

The proliferation of social media has transformed the Arabic-speaking world into a complex information ecosystem where constructive and destructive narratives compete. This duality is starkly represented by the concurrent rise of hate speech and hope speech, making their automatic detection paramount for content moderation and understanding online discourse (Mubarak et al., 2017). While early Arabic NLP efforts focused on general sentiment, the community has shifted towards more nuanced, high-impact tasks like hate speech detection.

The advent of large pre-trained Transformers (Devlin et al., 2019) has revolutionized this field, becoming the de facto standard. However, a fundamental architectural question remains for Arabic: do exclusively pre-trained Arabic-native models offer a performance advantage over large-scale multilingual models like XLM-RoBERTa (Conneau et al., 2020)? The latter may offer broader linguistic generalization, while the former might better capture language-specific nuances, dialects, and cultural contexts.

The MAHED 2025 shared task at ArabicNLP 2025 (Zaghouani et al., 2025) provides an ideal testbed to investigate this question. Its focus on the duality of hope and hate speech, alongside a complex emotion classification challenge, pushes beyond simple toxicity detection. In this paper, we present our systems for Subtask 1 and 2, systematically evaluating a diverse suite of Arabic-native and multilingual Transformer models to empirically answer this question. Our implementation is made publicly available to ensure reproducibility.[1]

The main contributions of our work:

- We present a systematic empirical comparison of seven distinct Transformer architectures, investigating the performance trade-offs between Arabic-native and multilingual encoders for nuanced affective computing.
- We developed robust systems for both subtasks, including a cascaded pipeline for Subtask 2 that explicitly models the hierarchical dependencies between offensive and hate speech detection, allowing for specialized classifier optimization.
- We establish a strong benchmark demonstrating the clear superiority of Arabic-native models, with our ARBERTv2-based system

[*] Authors contributed equally to this work.

[1] https://github.com/borhanitrash/ArabicNLP-EMNLP

achieving competitive performance. Our detailed error analysis further illuminates the specific challenges posed by semantic ambiguity and class imbalance in this domain.

2 Related Works

The automatic detection of nuanced affective states, including hate and hope speech, is a critical area of research in Arabic Natural Language Processing (NLP). Our work builds upon recent advancements in deep learning for sentiment and emotion analysis, particularly those leveraging Transformer-based architectures.

Recent efforts in Arabic affective computing highlight the success of pre-trained models. For instance, Cherrat et al. (2024) demonstrated the efficacy of AraBERT-based models for sentiment analysis across Standard Arabic and Moroccan dialect, showcasing their ability to capture complex linguistic features. Similarly, for Arabic tweet classification, Al-Onazi et al. (2023) developed a framework combining Deep Belief Networks with advanced hyperparameter optimization, while Elfaik et al. (2023) engineered a feature-fusion model using hybrid RNN-CNN architectures to tackle multi-label affect analysis. These studies affirm the power of deep learning for Arabic text but often focus on general sentiment or a broad spectrum of emotions.

This trend of applying sophisticated deep learning models extends to other languages and related tasks. Researchers have employed CNNs for detecting violent incitement in Urdu (Khan et al., 2024), hierarchical attention networks for depression detection from English tweets (Khafaga et al., 2023), and various hybrid architectures for emotion classification in Afan Oromo (Abdella and Sori, 2024). Furthermore, the field is advancing towards more complex methodologies, such as the tri-modal (text, audio, visual) graph neural networks for emotion recognition proposed by Al-Saadawi and Das (2024).

While these studies establish the effectiveness of Transformer models, a critical gap remains in the direct, empirical comparison of Arabic-native versus multilingual pre-training strategies for the complex, concurrent detection of hope, hate, and fine-grained emotions. Our work addresses this gap by leveraging the MAHED 2025 shared task as a rigorous testbed to provide a robust benchmark and a detailed analysis of model performance on this challenging domain.

Split	Instances	Unique Words	Total Words
Train	6,890	62,744	147,285
Validation	1,476	17,553	30,731
Test	1,477	17,891	31,492

Table 1: Dataset statistics for Subtask 1.

Split	Instances	Unique Words	Total Words
Train	5,960	45,015	115,279
Validation	1,277	13,726	25,346
Test	1,278	13,339	24,596

Table 2: Dataset statistics for Subtask 2.

3 Task and Dataset Description

We participated in the two text-based tracks of the MAHED 2025 shared task (Zaghouani et al., 2025), which provides a standardized framework to evaluate systems on challenging affective computing tasks in Arabic. We formalize the subtasks as follows:

Subtask 1: Hate and Hope Speech Classification. A three-way classification problem where the input is an Arabic text and the output is a single label from the set {`hate`, `hope`, `not_applicable`}. For example, a text translating to “All immigrants are thieves and criminals, they must be deported immediately” is labeled as `hate`.

Subtask 2: Emotion, Offensive, and Hate Detection. A multi-output classification problem with a hierarchical dependency. Given an Arabic text, the system must predict: (1) an emotion from a set of 12 labels (e.g., `anger`, `joy`); (2) a binary label indicating if the text is `offensive`; and (3) if offensive, a binary label indicating if it constitutes targeted `hate`. For instance, a text translating to “You donkey, why did you forget the keys?” is labeled as {`anger`, `yes`, `not_hate`}, distinguishing general offense from targeted hate.

The task organizers provided two annotated datasets (Zaghouani et al., 2024; Biswas and Zaghouani, 2025a,b) comprising text from online sources in both Modern Standard and dialectal Arabic. Dataset statistics are detailed in Table 1 and Table 2. The primary evaluation metric for both subtasks is the macro-averaged F1-score. For a more comprehensive analysis, we also report accuracy, and macro-averaged precision and recall.

4 Methodology

Our approach involves fine-tuning both multilingual and Arabic-native Transformer models (Vaswani et al., 2017), which excel at capturing the contextual cues necessary for nuanced hate and hope speech detection. We employed distinct strategies for the Hate and Hope Speech Classification (Figure 1) and the Emotion, Offensive, and Hate Detection (Figure 2) subtasks.

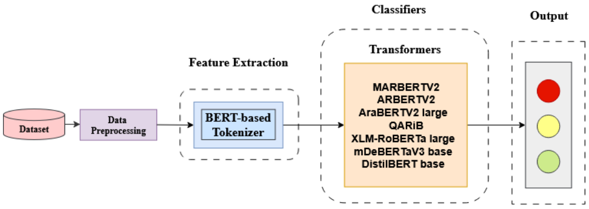


Figure 1: Schematic process for Hate and Hope Speech Classification.

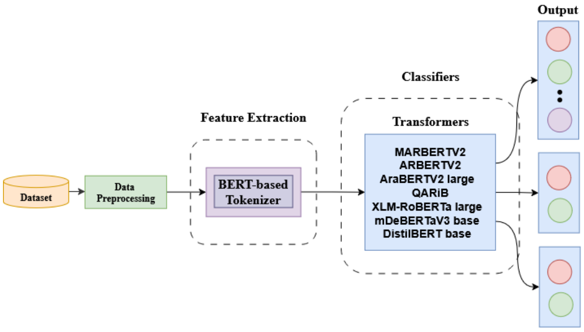


Figure 2: Schematic process for Emotion, Offensive, and Hate Detection.

4.1 Data Preprocessing

We implemented a unified text normalization pipeline for both subtasks prior to model-specific tokenization. The pipeline systematically removed URLs, user mentions, and hashtags, then normalized whitespace and filtered out non-Arabic characters. The cleaned text was subsequently processed using the AutoTokenizer corresponding to each pre-trained model. All input sequences were either padded or truncated to a fixed maximum length, generating `input_ids` and `attention_mask` tensors for model consumption.

4.2 Transformer-Based Models

Our selection of encoders was designed to evaluate a diverse range of pre-training objectives and linguistic specializations. Our model suite included Arabic-native encoders such as MARBERTV2 (UBC-NLP/MARBERTv2)[2] (Abdul-Mageed et al., 2021), ARBERTV2 (UBC-NLP/ARBERTv2)[3] (Abdul-Mageed et al., 2021), AraBERTV2 large (aubmindlab/bert-large-arabertv2)[4] (Antoun et al., 2020), and QARiB (ahmedabdelali/bert-base-qarib)[5] (Abdelali et al., 2021). These were complemented by powerful multilingual models, including XLM-RoBERTa large (FacebookAI/xlm-roberta-large)[6] (Conneau et al., 2020), mDeBERTaV3 base (microsoft/mdeberta-v3-base)[7] (He et al., 2021), and the computationally efficient DistilBERT base (distilbert/distilbert-base-multilingual-cased)[8] (Sanh et al., 2019). Each model was adapted for the downstream tasks as described below.

For Subtask 1, framed as a standard sequence classification problem, we fine-tuned each Transformer encoder by appending a sequence classification head. This head comprises a linear layer that takes the final hidden-state representation of the [CLS] token as input to produce logits for the three target classes. The entire fine-tuning process was managed using the Hugging Face Trainer API (Wolf et al., 2020), which optimized a standard Cross-Entropy Loss function. To prevent overfitting, we integrated an `EarlyStoppingCallback`, configured to monitor the macro F1-score on the official validation set and halt training after 3 epochs without improvement. The model checkpoint yielding the highest validation F1-score was preserved for the final test set evaluation.

In contrast, for Subtask 2, we addressed the task's explicit hierarchical dependency by designing a cascaded pipeline of three independently optimized classifiers. This modular design avoids the potential negative interference of joint multi-task optimization and allows each model to specialize. The pipeline consists of: an Emotion Classifier (12-class), an Offensive Classifier (binary), and a Hate Classifier (binary). The Hate classifier was trained exclusively on the subset of training data labeled as Offensive. During inference, test instances are processed in parallel by the Emotion and Offensive models; instances classified as Offensive are then routed to the Hate classifier for the final prediction. Each model in this pipeline was fine-tuned

[2] https://huggingface.co/UBC-NLP/MARBERTv2

[3] https://huggingface.co/UBC-NLP/ARBERTv2

[4] https://huggingface.co/aubmindlab/bert-large-arabertv2

[5] https://huggingface.co/ahmedabdelali/bert-base-qarib

[6] https://huggingface.co/FacebookAI/xlm-roberta-large

[7] https://huggingface.co/microsoft/mdeberta-v3-base

[8] https://huggingface.co/distilbert/distilbert-base-multilingual-cased

using a custom PyTorch loop, employing a class-weighted Cross-Entropy Loss to counteract severe label imbalance. Model selection for each of the three components was based on the highest macro F1-score achieved on the validation dataset.

All experiments were conducted with the AdamW optimizer (Loshchilov and Hutter, 2017) and utilized mixed-precision (FP16) training for computational efficiency. The specific hyperparameters for all models are detailed in Table 3.

Model	LR	WD	BS	EP
Subtask 1: Hate and Hope Classification				
MARBERTV2	2e-5	0.01	32	10
ARBERTV2	2e-5	0.01	32	10
AraBERTV2 large	1e-5	0.01	32	7
QARiB	2e-5	0.01	32	10
XLM-RoBERTa large	2e-5	0.01	16	10
mDeBERTaV3 base	2e-5	0.01	16	10
DistilBERT base	2e-5	0.01	16	10
Subtask 2: Emotion, Offensive, Hate				
MARBERTV2	2e-5	-	16	8
ARBERTV2	2e-5	-	16	8
AraBERTV2	2e-5	-	16	8
QARiB	2e-5	-	16	8
XLM-RoBERTa large	2e-5	-	16	8
mDeBERTaV3 base	2e-5	-	16	8
DistilBERT base	2e-5	-	16	8

Table 3: Hyperparameters used for fine-tuning. LR: Learning Rate, WD: Weight Decay, BS: Per-device Batch Size, EP: Max Epochs.

5 Result Analysis

This section presents the performance of our Transformer-based models on the MAHED 2025 shared task. All models were evaluated using the official metrics: accuracy, and macro-averaged precision, recall, and F1-score, with the macro F1-score serving as the primary metric for comparison. The comprehensive results for both subtasks are detailed in Table 4.

In Subtask 1, the Arabic-native models demonstrated a clear advantage over their multilingual counterparts. ARBERTv2 emerged as the top-performing system, achieving the highest macro F1-score of 0.6824 and the best accuracy of 0.6879. This strong performance is likely attributable to its pre-training on a large corpus of Arabic social media and web data, which aligns closely with the task's domain. Notably, MARBERTv2 secured the highest precision at 0.6824, indicating its proficiency in correctly identifying positive instances, albeit with a slightly lower overall F1-score. Other Arabic-specific models like QARiB and the multilingual mDeBERTaV3 base also delivered competitive results, underscoring the effectiveness of modern Transformer architectures. Conversely, AraBERTv2 large and DistilBERT base lagged behind, suggesting that either model scale or pre-training objective was less suited to this specific classification challenge.

Model	Accuracy	Precision	Recall	F1 Score
Subtask 1: Hate and Hope Speech Classification				
MARBERTv2	0.6804	**0.6824**	0.6562	0.6665
ARBERTv2	**0.6879**	0.6794	**0.6939**	**0.6824**
AraBERTv2 large	0.6269	0.6547	0.5714	0.5802
QARiB	0.6770	0.6664	0.6831	0.6738
XLM-RoBERTa large	0.6567	0.6514	0.6652	0.6554
mDeBERTaV3 base	0.6798	0.6716	0.6794	0.6729
DistilBERT base	0.6330	0.6258	0.6124	0.6110
Subtask 2: Emotion, Offensive, and Hate Detection				
MARBERTv2	0.7272	0.5040	0.5163	0.5078
ARBERTv2	0.7089	**0.5316**	**0.5257**	**0.5142**
AraBERTv2 large	0.6922	0.4765	0.4575	0.4593
QARiB	**0.7415**	0.5259	0.4943	0.4915
XLM-RoBERTa large	0.6896	0.4609	0.4564	0.4506
mDeBERTaV3 base	0.6907	0.4498	0.4619	0.4504
DistilBERT base	0.6468	0.3761	0.3801	0.3749

Table 4: Performance comparison of all evaluated models for Subtask 1 and Subtask 2. The best score in each column is highlighted in **bold**.

For the more complex, multi-output Subtask 2, ARBERTv2 once again demonstrated superior performance, leading across all macro-F1 (0.5142), precision (0.5316), and recall (0.5257) metrics. Its consistent success across both subtasks highlights the model's robustness and its ability to generalize well to related but distinct classification problems. MARBERTv2 followed closely with an F1-score of 0.5078. An interesting observation is the performance of QARiB, which achieved the highest accuracy (0.7415) but a lower F1-score of 0.4915. This discrepancy suggests the model may have excelled at predicting the majority classes (e.g., neutral emotion, no offensive) but struggled with the less frequent, yet critical, minority classes, reinforcing the importance of the macro F1-score as the primary evaluation metric in imbalanced scenarios.

Overall, our results indicate a distinct performance advantage for Arabic-native models pre-trained on diverse, user-generated content for both hate/hope speech detection and nuanced emotion classification. The performance gap between the two subtasks, with F1-scores being considerably lower in Subtask 2, underscores the inherent difficulty of the multi-output, hierarchically-dependent classification challenge. A detailed error analysis is provided in Appendix A.

6 Conclusion

In this paper, we presented our systems for the MAHED 2025 shared task, systematically evaluating Arabic-native and multilingual Transformer models on hope, hate, and emotion detection. Our findings consistently demonstrate the superiority of Arabic-native encoders, with our **ARBERTv2**-based system emerging as the top-performing model across both subtasks, achieving a macro F1-score of **0.682** (11th place) in Subtask 1 and **0.514** (5th place) in the more complex Subtask 2. The success of our cascaded classification pipeline in Subtask 2 underscores the value of modular models for hierarchical problems, though error analysis revealed persistent challenges in distinguishing nuanced emotional states and overcoming severe class imbalance, particularly for identifying targeted hate speech. Ultimately, this work contributes a robust benchmark comparing Arabic-native and multilingual models, affirming that domain- and language-specific pretraining remains crucial for tackling the subtleties of affective computing in Arabic social media.

Limitations

Our study is constrained by several limitations. Severe class imbalance, particularly in Subtask 2, significantly impacted our model's ability to detect the minority *hate* class, resulting in a conservative bias and a high number of false negatives. Our models also struggled with semantic nuance, often misclassifying subtle expressions of *hope* as neutral and confusing strong negative sentiment with targeted *hate* speech. The dataset, while valuable, may not fully capture the evolving nature of coded language across diverse Arabic dialects. Finally, our work was confined to the text modality, leaving the rich contextual information from the full multimodal task unexplored.

Acknowledgments

This work was supported by Southeast University, Bangladesh.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

Sufian Kedir Abdella and Worku Jifara Sori. 2024. Detection of emotions in afan oromo social media texts using deep learning method. *Ethiopian Journal of Science and Sustainable Development*, 11(1):70–84.

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.

Badriyya B Al-Onazi, Hassan Alshamrani, Fatimah Okleh Aldaajeh, Amira Sayed A Aziz, and Mohammed Rizwanullah. 2023. Modified seagull optimization with deep learning for affect classification in arabic tweets. *IEEE Access*, 11:98958–98968.

Hussein Farooq Tayeb Al-Saadawi and Resul Das. 2024. Ter-ca-wgnn: trimodel emotion recognition using cumulative attribute-weighted graph neural network. *Applied Sciences*, 14(6):2252.

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.

Md. Rafiul Biswas and Wajdi Zaghouani. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *CoRR*, abs/2505.11969.

Md. Rafiul Biswas and Wajdi Zaghouani. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *CoRR*, abs/2505.11959.

El Mehdi Cherrat, Hassan Ouahi, Abdellatif BEKKAR, and 1 others. 2024. Sentiment analysis from texts written in standard arabic and moroccan dialect based on deep learning approaches. *International Journal of Computing and Digital Systems*, 16(1):447–458.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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.

Hanane Elfaik and 1 others. 2023. Leveraging feature-level fusion representations and attentional bidirectional rnn-cnn deep models for arabic affect analysis on twitter. *Journal of King Saud University-Computer and Information Sciences*, 35(1):462–482.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

D Sami Khafaga, Maheshwari Auvdaiappan, K Deepa, Mohamed Abouhawwash, and F Khalid Karim. 2023. Deep learning for depression detection using twitter data. *Intelligent Automation & Soft Computing*, 36(2):1301–1313.

Muhammad Shahid Khan, Muhammad Shahid Iqbal Malik, and Aamer Nadeem. 2024. Detection of violence incitation expressions in urdu tweets using convolutional neural network. *Expert Systems with Applications*, 245:123174.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *First Workshop on Abusive Language Online 2017*, pages 52–56. Association for Computational Linguistics (ACL).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

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 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 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.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

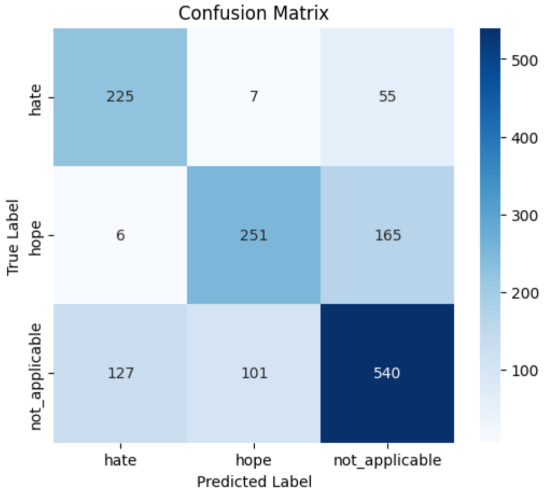
Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

A Error Analysis

We conducted a quantitative and qualitative error analysis of our best model, ARBERTv2, on the test set to understand its performance and limitations.

A.1 Quantitative Analysis

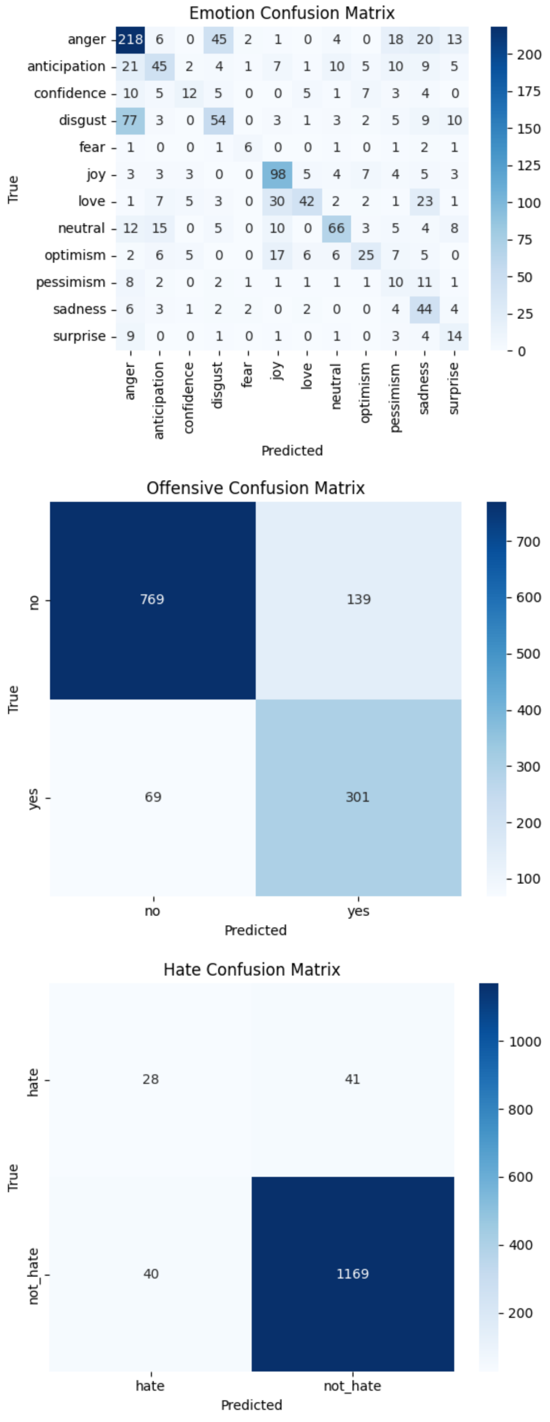
For Subtask 1, Figure 3 reveals key performance patterns. The model performs well on the not_applicable (540 true positives), hope (251), and hate (225) classes. However, it struggles with nuance, misclassifying 165 hope instances as not_applicable. Additionally, it misclassifies 127 not_applicable cases as hate, suggesting an oversensitivity to strong negative language.



Subtask 1

Figure 3: Confusion matrix of the proposed model (ARBERTv2) for Hate and Hope Speech Classification.

For Subtask 2, Figure 4 shows the challenges at each stage of our cascaded pipeline. In **Emotion Detection**, the model excels at high-frequency classes like anger (218) and joy (98) but struggles with fine-grained distinctions, often confusing optimism with neutral (25) or joy (17). In **Offensive Detection**, the model shows a conservative bias, missing 139 offensive instances (false negatives) while correctly identifying 301. Finally, severe data imbalance in the **Hate Detection** stage heavily impacts performance; the model misclassifies 41 hate cases as not_hate while correctly identifying only 28, showing its difficulty in distinguishing targeted hate from general offensiveness.



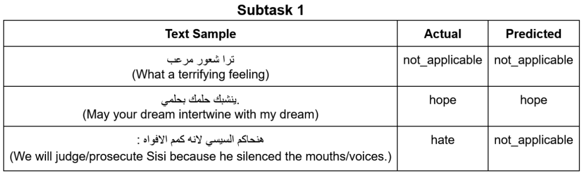
Subtask 2

Figure 4: Confusion matrices of the proposed model (ARBERTv2) for Emotion, Offensive, and Hate Detection.

A.2 Qualitative Analysis

Qualitative analysis of misclassifications reveals further limitations of ARBERTv2. For Subtask 1 (Figure 5), a politically charged text implying hostility was misclassified as `not_applicable` instead of `hate`, highlighting the model's difficulty with implicit threats that lack explicit slurs. For Subtask 2 (Figure 6), a text containing an expletive was mislabeled as `neutral` instead of `anger`. The formal phrasing seemingly overrode the informal expletive, highlighting challenges with mixed-tone sentences.

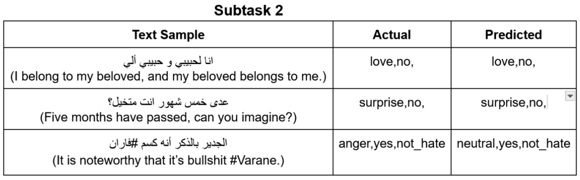
These observations confirm the system's primary weaknesses: handling nuanced language, distinguishing related emotions, and overcoming data imbalance, especially for targeted hate speech detection.



Subtask 1

Text Sample	Actual	Predicted
ترا شعور مرعب (What a terrifying feeling)	not_applicable	not_applicable
ينشبك حلمك بحلمي. (May your dream intertwine with my dream)	hope	hope
هنحاكم السيسي لانه كمم الافواه : (We will judge/prosecute Sisi because he silenced the mouths/voices.)	hate	not_applicable

Figure 5: Few examples of predictions produced by the proposed ARBERTv2 model on Subtask 1.



Subtask 2

Text Sample	Actual	Predicted
انا لحبيبي و حبيبي ألي (I belong to my beloved, and my beloved belongs to me.)	love,no,	love,no,
عدى خمس شهور انت متخيل؟ (Five months have passed, can you imagine?)	surprise,no,	surprise,no,
الجدير بالذكر أنه كسم #فاران (It is noteworthy that it's bullshit #Varane.)	anger,yes,not_hate	neutral,yes,not_hate

Figure 6: Few examples of predictions produced by the proposed ARBERTv2 model on Subtask 2.

Baoflowin502 at MAHED Shared Task: Text-based Hate and Hope Speech Classification

Nguyen Minh Bao
University of Information Technology
23520123@gm.uit.edu.vn

Dang Van Thin
University of Information Technology

Abstract

This paper presents Arabic hate and hope speech classification using pre-trained language models and advanced data augmentation techniques. We evaluate multiple Arabic BERT variants on 6,889 Arabic text samples labeled as hate speech, hope speech, or not-applicable. Data augmentation through back-translation and LLM-based data generation using few-shot prompting significantly improves performance across all models. We establish strong baselines using ensemble bagging XGBoost alongside traditional machine learning approaches. CAMeLBERT with data augmentation achieves the best Macro-F1 of 0.6868, demonstrating the effectiveness of Arabic-specific models combined with modern augmentation strategies for hate speech detection speech detection.

1 Introduction

The proliferation of social media has transformed communication while significantly accelerating the spread of hate speech and harmful content. In this competition (Bao, 2025) (Zaghouani et al., 2025) (Zaghouani et al., 2024), we tackle the Arabic hate–hope–neutral speech classification problem — a challenging NLP task due to Arabic's morphological richness, diverse dialects, and right-to-left script. These linguistic complexities, along with subtle cultural and contextual cues, make model development more difficult than for high-resource languages like English (Wahdan et al., 2024; Elnagar et al., 2020).

We conduct a systematic evaluation of three leading Arabic-specific BERT variants — AraBERTv2 (Antoun et al.), CAMeLBERT (Inoue et al., 2021), and MARBERT (Abdul-Mageed et al., 2021) — chosen for their complementary strengths in handling formal, morphologically complex, and multi-dialectal Arabic text. To mitigate data scarcity, we adopt a dual augmentation strategy: multi-hop back-translation to generate natural paraphrases and LLM-based few-shot prompting (Kim et al., 2024) to produce contextually coherent synthetic examples. For comparison, we also implement strong traditional baselines using ensemble bagging XGBoost on contextual embeddings.

Our contributions include: a systematic benchmark of high-performing Arabic BERT variants for hate–hope speech classification, a tailored augmentation pipeline for Arabic text that combines cross-lingual and generative approaches, and the development of a hard voting ensemble method that leverages the complementary abilities of multiple models, achieving consistent performance improvements and advancing Arabic NLP for content moderation applications.

2 Related Work

2.1 Text Preprocessing for Arabic Social Media

Effective preprocessing involves converting emojis to textual representations using comprehensive emoji-to-text dictionaries to preserve emotional context essential for hate and hope sentiment analysis. Text normalization through tokenization ensures consistent input representation, handling variations in spelling, punctuation, and formatting commonly found in social media posts. Additional preprocessing includes URL removal, mention cleaning, and Arabic text standardization to optimize model performance while preserving linguistically relevant information for classification tasks.

2.2 Data Augmentation Strategies

Data augmentation addresses low-resource scenarios through two sophisticated techniques. Back-translation leverages machine translation systems to generate paraphrases by translating text through intermediate languages and back to the

Proceedings of The Third Arabic Natural Language Processing Conference, pages 665–669
November 8-9, 2025 ©2025 Association for Computational Linguistics

source language, creating diverse training examples while preserving semantic meaning. Large language model-based data generation using few-shot prompting provides contextually appropriate training examples by leveraging in-context learning capabilities with representative sample prompts. The combination of back-translation and LLM-based few-shot generation provides complementary benefits, addressing different aspects of data scarcity while ensuring high-quality augmented datasets.

2.3 Arabic Language Model Selection

Through comprehensive literature survey, we selected three prominent Arabic-specific BERT variants demonstrating superior performance in Arabic NLP tasks: AraBERTv2 (comprehensive Arabic BERT pre-trained on large-scale Arabic corpora), CAMeLBERT (advanced model with optimized architecture for Arabic morphological features), and MARBERT (multi-dialectal Arabic specialist for diverse text processing). This focused selection ensures robust evaluation of the most established Arabic language models for hate and hope speech classification tasks.

3 Methodology

3.1 Dataset Description

Our experimental dataset comprises 6,889 Arabic text samples systematically collected from diverse social media platforms including Twitter, Facebook, and regional Arabic forums. The dataset encompasses three distinct classification categories: **Hate Speech** samples (2,296 instances, 33.3%) containing explicit or implicit expressions of hatred and discrimination targeting individuals or groups; **Hope Speech** samples (2,301 instances, 33.4%) promoting positive values, social inclusion, and constructive dialogue; and **Not-applicable** samples (2,292 instances, 33.3%) representing neutral content that does not clearly fall into either category. The balanced class distribution provides a solid methodological foundation for robust model training, while geographic diversity across different Arab-speaking regions ensures linguistic representativeness.

3.2 Data Preprocessing Pipeline

We implement a comprehensive preprocessing pipeline specifically tailored for Arabic social media text. The process includes emoji replacement using extensive multilingual dictionaries containing over 3,000 mappings to preserve emotional context crucial for sentiment classification; URL detection and removal via robust regular expressions while preserving adjacent contextual information; Arabic text normalization addressing script-specific challenges including variant letter forms and punctuation standardization; tokenization using NLTK's Arabic-specific algorithms enhanced with custom rules for morphological patterns; and systematic mention removal to reduce person-specific bias while maintaining relevant surrounding context.

3.3 Data Augmentation Strategies

We employ three complementary augmentation strategies with a 50% augmentation ratio to balance dataset expansion with computational efficiency:

3.3.1 Multi-hop Back-Translation

Multi-step translation process using Google Translate API following Arabic → English → French → Arabic sequence. This approach introduces natural linguistic variations through different language typologies while preserving semantic content. Quality control includes automatic filtering of translation artifacts and semantic similarity verification using multilingual embeddings.

3.3.2 LLM-based Few-Shot Data Generation

Leveraging GPT-4 with carefully designed prompting strategies providing 3-5 representative examples per class. The methodology includes explicit instructions for maintaining dialectal authenticity, appropriate emotional intensity, and realistic social media communication patterns. Quality assurance involves automated toxicity filtering and semantic coherence verification.

3.3.3 Controlled Lexical Substitution

Systematic replacement using Arabic WordNet and curated synonym dictionaries, selectively targeting non-key terms to introduce lexical diversity without altering core semantic meaning. The process incorporates POS tagging and NER to preserve proper nouns and category-specific terminology essential for classification accuracy.

4 Model Architecture and Experimental Setup

4.1 Pre-trained Language Models

We evaluate three prominent Arabic-specific BERT variants based on comprehensive literature survey:

AraBERTv2 (comprehensive Arabic BERT pretrained on large-scale Arabic corpora), CAMeLBERT (advanced model with optimized architecture for Arabic morphological features), and MARBERT (multi-dialectal Arabic specialist for diverse text processing). This focused selection ensures robust evaluation of the most established Arabic language models for hate and hope speech classification tasks.

4.2 Hard Voting Ensemble Method

To leverage the complementary strengths of individual Arabic BERT models, we implement a hard voting ensemble combining predictions from AraBERTv2, CAMeLBERT, and MARBERT. In this ensemble approach, each model independently processes the input text and generates predictions for the three classes (Hate, Hope, Not-applicable). The final prediction is determined through majority voting, where the class receiving the most votes across the three models is selected as the ensemble output. In cases of tie situations, we implement a confidence-based tie-breaking mechanism using the model with the highest prediction probability. This hard voting strategy capitalizes on the diverse strengths of each Arabic BERT variant: AraBERTv2's comprehensive Arabic coverage, CAMeLBERT's morphological optimization, and MARBERT's dialectal expertise, potentially improving overall classification robustness and accuracy.

4.3 Traditional Machine Learning Baselines

For a comprehensive performance comparison, we establish strong traditional machine learning baselines. Specifically, we implement an ensemble bagging XGBoost classifier that operates on vector embeddings extracted from the AraBERTv2 model. By combining the representational power of contextualized AraBERTv2 embeddings with XGBoost's gradient boosting capabilities, this setup effectively captures both semantic and lexical patterns present in the Arabic text. The use of bagging further enhances robustness by reducing variance and mitigating overfitting, thus providing a solid benchmark against which transformer-based approaches can be evaluated. In addition to the gradient boosting baseline, we also evaluate a Multi-Layer Perceptron (MLP) with a single hidden layer.

4.4 Experimental Configuration

Our setup ensures rigorous evaluation through Stratified K-Fold cross-validation to maintain balanced class representation, the Optuna framework (Akiba et al., 2019) for Bayesian hyperparameter optimization, and macro-averaged Macro-F1 as the primary metric for balanced evaluation across classes. All transformer models are trained with GPU acceleration on an NVIDIA Tesla P100, ensuring efficient experimentation and reproducible results.

5 Results and Analysis

5.1 Baseline Performance

Baseline experiments without augmentation reveal important insights into Arabic hate and hope speech classification challenges. Ensemble bagging XGBoost achieved 0.626 Macro-F1, demonstrating traditional gradient boosting effectiveness with AraBERTv2 embeddings. MLP reached 0.616, showing marginal improvement despite neural architecture. Among individual Arabic BERT models, AraBERTv2 obtained 0.623, CAMeLBERT achieved 0.647, and MARBERT reached 0.639, representing comparable baseline performance with modest gains over traditional approaches.

These results reveal task characteristics: close performance between traditional ML and transformers suggests three-class classification challenges stem from inherent difficulties distinguishing between categories rather than sophisticated feature representations. Moderate Macro-F1s indicate significant challenges likely due to subtle distinctions and cultural context requirements.

Model	**Macro-F1**
Ensemble Bagging XGBoost	0.626
MLP	0.616
AraBERTv2	0.623
CAMeLBERT	0.647
MARBERT	0.639

Table 1: Baseline results without data augmentation (5-fold CV validation); Metric: Macro-F1

5.2 Impact of Data Augmentation and Ensemble Methods

Data augmentation yields substantial improvements across all models. AraBERTv2 improved to 0.665, MARBERT achieved 0.652, and CAMeLBERT reached 0.679. The hard voting ensemble combining all three Arabic BERT variants achieved 0.689, representing the highest performance and demonstrating the effectiveness of leveraging complementary model strengths.

Consistent improvements validate our ensemble augmentation strategy combining back-translation and LLM-based few-shot generation. This approach addresses different data scarcity aspects while maintaining semantic properties essential for classification tasks.

Model	Macro-F1
Individual BERT Models	
AraBERTv2	0.665
MARBERT	0.652
CAMeLBERT	0.679
Ensemble Methods	
Hard Voting Ensemble	**0.689**
Ensemble Bagging XGBoost	0.656

Table 2: Results with data augmentation and ensemble methods (5-fold CV validation); Metric: Macro-F1

5.3 Model Analysis

The hard voting ensemble's superior performance (Macro-F1=0.689) demonstrates the value of combining diverse Arabic BERT variants, leveraging AraBERTv2's comprehensive coverage, CAMeLBERT's morphological optimization, and MARBERT's dialectal expertise. Among individual models, CAMeLBERT's strong performance (Macro-F1=0.679) stems from its optimized architecture for Arabic linguistic features.

Traditional ML competitive performance relative to individual transformers suggests primary challenges relate to dataset size and inherent task complexity rather than model capacity limitations, with practical implications for resource-limited scenarios.

6 Discussion

6.1 Task Complexity and Ensemble Benefits

The moderate Macro-F1 achieved by individual models (0.652–0.679) and the ensemble approach (0.687) underscore the inherent difficulty of three-class hate/hope speech classification, driven by subjective label boundaries, cultural context dependencies, and subtle linguistic cues. The superior results of the hard voting ensemble indicate that integrating diverse Arabic BERT variants can effectively leverage their complementary strengths to better handle these challenges.

6.2 Practical Implications

Ensemble data augmentation combining back-translation and LLM-based few-shot generation should be standard practice for Arabic datasets. Hard voting ensemble superiority reinforces the value of leveraging multiple Arabic-specific models over single architectures, with practical benefits for content moderation systems.

6.3 Future Directions

Promising directions include larger datasets with broader dialectal coverage, soft voting and weighted ensemble methods, multi-task learning approaches, and explainability research for ensemble decision-making processes.

7 Conclusion

We present an effective Arabic hate and hope speech classification approach using a hard voting ensemble of AraBERTv2, CAMeLBERT, and MARBERT with multi-hop back-translation and LLM-based few-shot augmentation, achieving an Macro-F1 of 0.689 in this competition. This work systematically evaluates leading Arabic models and validates that combining complementary architectures with advanced augmentation significantly boosts performance, providing a solid foundation for practical, culturally aware Arabic content moderation systems.

Acknowledgments

We thank anonymous reviewers for valuable feedback and the University of Information Technology for computational resources supporting this 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.

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 & data mining*, pages 2623–2631.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Nguyen Minh Bao. 2025. baoflowin502 at MarsadLab: Baoflowin502 at mahed2025: Text-based hate and hope speech classification. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Ashraf Elnagar, Ridhwan Al-Debsi, and Omar Einea. 2020. Arabic text classification using deep learning models. *Information Processing & Management*, 57(1):102121.

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.

Jae Myung Kim, Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. 2024. Datadream: Few-shot guided dataset generation. In *European Conference on Computer Vision*, pages 252–268. Springer.

Ahlam Wahdan, Mostafa Al-Emran, and Khaled Shaalan. 2024. A systematic review of arabic text classification: areas, applications, and future directions. *Soft Computing*, 28(2):1545–1566.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

AyahVerse at MAHED Shared Task: Fine-Tuning ArabicBERT with Preprocessing for Hope and Hate Detection

Ibad-ur-Rehman Rashid, Muhammad Hashir Khalil
Government Post Graduate College, Mansehra, Pakistan
ibad@gcm.edu.pk, hashirkhalil3@gmail.com

Abstract

We participated in Subtask 1 of the MAHED Shared Task 2025, which focuses on detecting hope, hate, and not applicable labels in Arabic content. In this work, we tested a multiclass classifier for hope, hate, and not applicable label detection from a dataset provided by organizers as Subtask 1. We approach the task by two methods. The first one is a fine-tuned Arabic model, ArabicBERT, on a multiclass classification task. The second one is a two-step stacked architecture. Both of them include a dedicated pipeline for specific Arabic preprocessing with different techniques. Official results are 0.38 F1 score on the validation set and 0.47 on the test set with a single multiclass classifier. Post-submission improvements resulted in macro-F1 scores of 0.60(validation) and 0.63(test) for the single classifier, and 0.59(validation) and 0.91(test) for the stacked classifier.

1 Introduction

The MAHED Shared Task 2025 (Subtask 1)(Zaghouani et al., 2025) focuses on detecting hope and hate emotions in Arabic text from social media content and tweets about Middle East conflict. The dataset(Zaghouani and Biswas, 2025a) is annotated with the labels "hope" "hate" and "not applicable". Training on this task by participants contributes to new methodologies in Arabic NLP, especially in the classification of hate and hope speech and multimodal content detection for understanding online discourse and promoting positive engagement in social media communities. During the task we discover many challenges, including overfitting, underfitting, class imbalance, and label inconsistencies in a few cases. Our code is available at GitHub[1].

We discovered that proper Arabic preprocessing significantly improves performance. Undersampling and oversampling led to overfitting, and adjusted weights resulted in a performance increase in both validation and test sets. The difference between a single multiclass classifier and a stacked classifier is minimal, however removing emojis in stack classifier configuration resulted in a notable improvement, reaching 0.91 F1 score on the official test set.

The preprocessing pipeline consists of some general language preprocessing techniques like URL and hashtag removal, as well as some specific Arabic language preprocessing with different techniques like handling class imbalance, diacritization, and Arabic letter normalization.

We focused on two model architectures. The first one is a fine-tuned multiclass ArabicBERT(Safaya et al., 2020)[2] for predicting hate, hope, or not applicable labels in the text. The second model architecture consists of two binary fine-tuned 'ArabicBERT' classifiers for detecting hate and hope speech, and this layer is stacked with a final logistic regression meta-classifier.

On the official leaderboard, our system ranked 25th out of 25 teams, achieving a macro-F1 of 0.48 on the test set. Post-submission results increase F1 scores to around 0.60 in the validation set, and 0.91 in the test set in different configurations.

2 Background

2.1 Task setup

The shared task required participants to classify Arabic text into one of the classes:

- *hope* hopeful messages
- *hate* hateful or abusive messages

[1] https://github.com/Ebad-urRehman/MAHED_2025_subtask1_hate_and_hope/

[2] https://huggingface.co/asafaya/bert-base-arabic

Proceedings of The Third Arabic Natural Language Processing Conference, pages 670–676
November 8-9, 2025 ©2025 Association for Computational Linguistics

- *not_applicable* neutral content

Example

- أَنَا مُتَحَمِّسٌ لِلشُّعُورِ بِالرُّوحِ الْخَاصَّةِ الَّتِي سَتَجْلِبُهَا إِلَى مَنْزِلِنَا
  → *hope*
- أنتم لا تستحقون الاحترام
  → *hate*
- تحب تقول ايه لتميم بن موزه يا جبان ما انت جنست ناس من السنغال للدفاع عنك
  → *not_applicable*

2.2 Dataset

We used the provided MAHED 2025 dataset (Zaghouani and Biswas, 2025a), consisting of Arabic social media posts gathered considering the linguistic diversity and dialect variations. The dataset is labeled for hope, hate, and not applicable categories, and it contains train, test, and validation splits D. The split used for the test set during training is 0.2. We use only training data for training of all models.

The full dataset size is 9843, with 6890 for training, 1476 for validation, and 1477 for testing.

Datasets for Subtask 2 are (Zaghouani and Biswas, 2025b) and (Zaghouani et al., 2024) which are not used in this work.

2.3 Track

We participated in Subtask 1 of the MAHED Shared Task 2025.

2.4 Related Works

Recent studies on Arabic hate speech, including (Althobaiti, 2022) provide a comparison between the BERT-based approach and two machine learning techniques, demonstrating that BERT-based models are more effective. They also experimented with incorporating sentiment information along with text into the BERT model and converting emojis to textual descriptions. While sentiment features slightly improved performance, the effect of emoji descriptions varied depending on class distribution.

(Almaliki et al., 2023) is a benchmark model for Arabic offensive language detection, which is classified into three classes: normal, abuse, and hate speech. Another study, (Aldjanabi et al., 2021) Using a Cross-Corpora Multi-Task Learning Model,' trained a model on a wide variety of datasets in multiple tasks; their model was fine-tuned on the MarBERT (Abdul-Mageed et al., 2021) Arabic model. Similarly, 'BERT-CNN for Offensive Speech Identification in Social Media' combines CNN with BERT and demonstrates the effectiveness of the ArabicBERT model when combined with CNN.

A multi-task learning strategy was more recently experimented with by (Abdelsamie et al., 2026) to address dialectal variations in Arabic hate speech detection. Their model captures the distinctive features of each of the five Arabic dialects (Egyptian, Levant, Saudi, Algerian, and Gulf) while leveraging shared knowledge across them. With remarkable F1 scores of 0.98, 0.84, 0.85, 0.76, and 0.80 for the corresponding dialects, it outperformed single-task models by about 14%.

In contrast to these studies, our system differs in the datasets used and the training approaches we employed. We experimented with different levels of preprocessing, including Arabic letter normalization, diacritics, and tatweel removal. The approaches we explored were two main strategies: a single multiclass classifier and a stacked binary ensemble of classifiers with two approaches. In the ensemble, one variant includes all 'not applicable' labels in both binary classifiers, while another variant splits 'not applicable' labels into two subsets to use separately with both binary classifiers.

3 System Overview

We choose ArabicBERT because it is one of the high-performing models of nlp arabic as per (Alammary, 2022). We aim to test it for multiclass classification with a single classifier as well as a stacked multilayer architecture. We trained and tested our model on provided datasets only.

3.1 Preprocessing

At first we implemented simple preprocessing techniques like URL, hashtag, handle, and stopword removal.

In later versions we included some specific Arabic preprocessing techniques including:

1. Mapped emojis to Arabic text equivalents using the defined ‘emoji to text‘ dictionary.

2. Character normalization e.g., أ, إ, آ → ا; ى → ي.

3. Diacritics and tatweel removal for a unified formatted dataset and noise removal for better model understanding.

4. We also implemented some general preprocessing techniques like URL, hashtag, and handle removals and whitespace normalization, without stopword removal, as they carry context and meaning in Arabic.

5. For handling class imbalance, we used different techniques like undersampling, oversampling, and adjusting class weights.

3.2 Model

We tested different model configurations. Two of the main architectures are (i) a single multiclass classifier and stacked binary ensemble with a meta classifier.

3.2.1 For Single Multiclass Classifier:

Our system follows a preprocess → tokenize → classify → evaluate pipeline.

At the start, we experimented with a deeper classification head consisting of an additional fully connected layer of size 256 with a ReLU activation function on top of ArabicBERT. This 256-dimensional layer was connected to the final output layer, producing three logits, using the same general preprocessing pipeline. However, this design showed poor generalization. We then tried a simpler classifier where ArabicBERT was directly connected to a linear layer producing three logits, followed by dropout with improved preprocessing. This setup gave better performance and stability on both training and validation and was therefore chosen as our final classifier design.

After this we tested the selected model with different levels of preprocessing and class imbalance handling techniques like undersampling, oversampling, and adjusting class weights for loss calculation (see Appendix E). In addition to this multiclass approach, we also designed and tested a stacked binary ensemble architecture.

3.2.2 For stacked binary ensemble:

Our system follows a preprocess → tokenize → classify → ensemble → evaluate pipeline.

It is a two-step stacked architecture, where one ArabicBERT model was trained to classify hope vs. not applicable and another to classify hate vs. not applicable, with their probability outputs fed into a logistic regression meta-classifier for final prediction. The binary ArabicBERT model configurations are kept the same as the single multiclass classifier. Like the single multiclass classifier, we also tested this for different levels of preprocessing and class imbalancing handling techniques.

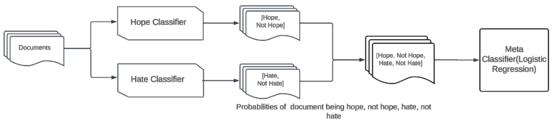

Documents
Hope Classifier
Hate Classifier
[Hope, Not Hope]
[Hate, Not Hate]
[Hope, Not Hope, Hate, Not Hate]
Meta Classifier(Logistic Regression)
Probabilities of document being hope, not hope, hate, not hate

Figure 1: Meta Classifier Architecture

In another variant, as shown in the figure below, we applied an additional preprocessing step where the not applicable class was divided into two equal parts. One part was used alongside the hope examples, and the other part was paired with the hate examples for training. We tried this because binary classifiers (hope and hate) are underperforming on hate and hope classes due to more examples of not applicable in dataset. Binary class performance on this new architecture improves; however, performance of the meta-classifier in both stacked architectures yields close results.

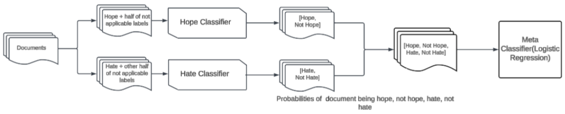

Documents
Hope + half of not applicable labels
Hate + other half of not applicable labels
Hope Classifier
Hate Classifier
[Hope, Not Hope]
[Hate, Not Hate]
[Hope, Not Hope, Hate, Not Hate]
Meta Classifier(Logistic Regression)
Probabilities of document being hope, not hope, hate, not hate

Figure 2: Meta Classifier with not applicable labels split

3.3 Challenges

The MAHED 2025 hope and hate text classification dataset is highly imbalanced. We explored several strategies, including oversampling, undersampling, and adjusting class weights. These approaches lead to more overfitting and underfitting and eventually a low F1 score for validation and test sets, except for adjusting class weights that gives an increase in F1 score.

However, the best results were achieved by performing specific Arabic preprocessing, without applying class imbalance techniques. Increasing the number of training epochs from 3 to 8 slightly improves the performance.

Earlier we tried to test without stopwords, but later we decided to retain them, and this is also a reason for improved performance in later experiments.

4 Experimental Setup

We fine-tuned ArabicBERT[3] using the Transformers library. Our model used the AdamW optimizer with the CrossEntropyLoss function, going through a training of 8 epochs. The max sequence length for sentences is 128, the single batch size is 16, and the learning rate is 2×10^{-5}.

We trained, validated, and tested our model using the official datasets. During training, 20% of the data was reserved for testing. Training of all models was performed only on `train.csv`.

Our implementation used Python 3.13, PyTorch, HuggingFace Transformers, scikit-learn, NLTK, pandas. Experiments are conducted on Google colab GPUs T4, L4 and A100. Consuming approximately 70 compute units on training, and testing.

5 Results

5.1 Official Results from scoring files

Official results from the scoring files show low scores, because model is underfitting due to exclusion of proper arabic preprocessing like diacritization, Arabic letter normalization, and converting emojis to arabic text. Another reasons of low F1 scores are custom layer on top of bert classifier, and less number of epochs. We also have not experimented with stacked classifier at that time. The official results of the scoring files are shown in Table 1.

Metrics (Macro)	F1	Accuracy	Precision	Recall
Validation File	0.376	0.649	0.376	0.377
Test File	0.465	0.624	0.458	0.474

Table 1: Official results.

5.2 Post Submission Results

F1 scores significantly improves in post submission experiments, because of specific arabic preprocessing pipeline and increased number of epochs.

Metrics (Macro)	F1	Accuracy	Precision	Recall
Validation File	0.603	0.621	0.596	0.612
Test File	0.608	0.632	0.628	0.594

Table 2: Post-submission performance (Macro metrics) of Single Multiclass Classifier with weight adjustments.

Metrics (Macro)	F1	Accuracy	Precision	Recall
Validation File	0.60	0.61	0.59	0.61
Test File	0.63	0.63	0.62	0.64

Table 3: Post-submission performance (Macro metrics) of Stacked Binary Ensemble Classifier.(with emojis replaced with arabic text)

5.3 Analysis

5.3.1 Analysis of Single Multiclass Classifier under different strategies

Our single multiclass classifier achieves an F1 score of 0.71 on training, 0.57 on validation, and 0.60 on test sets. With adjusted weights, our multiclass classifier achieves 0.98 on training, 0.60 on validation, and 0.63 on testing. With oversampling, we observe overfitting because duplicate examples may make the model memorize some examples instead of generalizing. With undersampling, too, we observe overfitting because of missing examples in training data, which makes the model perform poorly on test data.

Model	Train/Test	Validation	Test
No Strategy	0.717	0.578	0.608
With Oversampling	0.984	0.252	0.244
With Undersampling	0.809	0.236	0.231
With Adjusted weights	0.986	0.603	0.635

Table 4: Macro-F1 comparison across different training strategies for the Single Multiclass Classifier details in Appendix Table 6. A

With emojis removed instead of being replaced with Arabic text and no class imbalance technique applied, the model gives F1 scores of 0.60 and 0.62 for validation and test files, respectively. While removing emojis and adjusted weights gives an F1 score of 0.58 on validation and 0.64 on testing, details in Appendix Table 7. A

5.3.2 Analysis of Stacked Binary Ensemble Classifier under different strategies

Our two-layer stacked binary ensemble classifier achieves an F1 score of 0.60 on the validation set and 0.63 on the test set when emojis are replaced with Arabic words. When emojis are completely removed, the F1 score changes to 0.59 on validation and 0.91 on the test set.

In the second variant of the stacked binary ensemble, which includes an additional preprocessing step that splits the not applicable label into two subsets, the F1 score is 0.58 for validation and 0.63 for the test set. Excluding emojis in this configu-

[3] https://huggingface.co/asafaya/bert-base-arabic

ration results in an F1 score of 0.59 on validation and 0.65 on the test set.

Metrics (Macro)	F1	Accuracy	Precision	Recall
Training Performance	0.90	0.61	0.89	0.90
Validation File	0.60	0.61	0.59	0.61
Val (without emojis)	0.59	0.61	0.59	0.59
Test File	0.63	0.63	0.62	0.64
Test (without emojis)	0.91	0.91	0.92	0.91

Table 5: Post-submission performance of the Stacked Binary Ensemble classifier, details in Appendix Table 8. A

System Error Examples. The increase in F1-scores due to emoji removal in the test set might be due to sarcasm examples where a laughing emoji is used, but the overall text is hate. In such cases, removing emojis instead of converting them into Arabic equivalent words helps the models understanding.

For example, the translation of [laughing] emoji in the dictionary is ضحك (laughing), which gives a hopeful sentiment. However, in the dataset it appears frequently in hate and not-applicable examples as shown in Appendix. B

Some annotation mistakes also contributed to poor model understanding and thus lower performance as shown in Appendix. C

Our system sometimes overfits, especially with oversampling or deeper classification heads. This causes high training F1 scores but poor performance in validation / test sets. The performance of the system can be improved by better data quality, proper preprocessing, and the use of a suitable class imbalance handling technique.

6 Conclusion

In this work, we aimed to classify Arabic social media posts into hope, hate, and not applicable categories as part of MAHED Shared Task 2025 Subtask 1. We developed a multiclass classifier based on the Arabic model ArabicBERT, fine-tuned on the competition dataset, and achieved an F1 score of 0.60 and 0.63 on the given validation and test datasets. With our other approach, we tested stacked binary ensemble models and achieved F1 scores of 0.59 and 0.91 on validation and test sets. Specific Arabic preprocessing choices, like skipping stopword removal, normalizing Arabic letters, removing diacritics, and tatweel, resulted in improvement of the F1 score. The adjusted class weights technique for handling class imbalance performs better as compared to other techniques like oversampling and undersampling.

Acknowledgments

We would like to thank contributors and co-authors of this task, including Muhammad Hashir Khalil, Junaid Hussain, Syed Saqlain Gillani, and Shahid Khan, for providing assistance in writing this paper. We also thank the organizers of the MAHED Shared Task 2025 for providing the dataset and evaluation platform. We sincerely thank the anonymous reviewers for their insightful comments and constructive feedback, which greatly improved the quality of this work. Furthermore, we also acknowledge Hugging Face for open-sourcing ArabicBERT and Google Colab for providing the GPU resources used in training and evaluation.

References

Mahmoud Mohamed Abdelsamie, Shahira Shaaban Azab, and Hesham A. Hefny. 2026. The dialects gap: A multi-task learning approach for enhancing hate speech detection in arabic dialects. *Expert Systems with Applications*, 295:128584.

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.

Ali Saleh Alammary. 2022. Bert models for arabic text classification: A systematic review. *Applied Sciences*, 12(11).

Wassen Aldjanabi, Abdelghani Dahou, Mohammed A. A. Al-qaness, Mohamed Abd Elaziz, Ahmed Mohamed Helmi, and Robertas Damaševičius. 2021. Arabic offensive and hate speech detection using a cross-corpora multitask learning model. *Informatics*, 8(4).

Malik Almaliki, Abdulqader M. Almars, Ibrahim Gad, and El-Sayed Atlam. 2023. Abmm: Arabic bert-mini model for hate-speech detection on social media. *Electronics*, 12(4).

Maha Jarallah Althobaiti. 2022. Bert-based approach to arabic hate speech and offensive language detection in twitter: Exploiting emojis and sentiment analysis. *International Journal of Advanced Computer Science and Applications*, 13(5).

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054--2059, Barcelona (online). International Committee for Computational Linguistics.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044--15055.

A Tables

B Emoji Effects Example

[laughing emojis]

انا خايفه ادخل تويتر مو بس حسابك

"I'm afraid to log into Twitter, not just your account."
Label: Not Applicable

نعوضكم بأيش يعني قم انقلع بس يالعرا كي [laughing emojis]
"How do we compensate you? [laughing emojis] Just get lost, you brat."
Label: Hate

C Incorrect Annotation Example

علشان انتى بومه وفقريه وهنفوز بأداء ونتيجه ان شاء الله الاهلى بيلعب كرة حلوة مع الفرق اللى بتلعب كرة وبعدين انتى تعرفي عن الكرة ايه غير انها مدورة [laughing emojis]
"Because you are an owl and a poor girl, and we will win with performance and results, God willing. Al-Ahly plays good football with teams that play football. And what do you know about football other than that it is round? [laughing emojis]"
Annotated Label: Hope
Correct Label: Hate
Model Predicted: Hope

السعوديه الثالثه في السمنه تباً لكم مرهلين وش قعدني معكم
"Saudi Arabia is third in obesity. Damn you, you idiots. What made me stay with you?"
Annotated Label: Hope
Correct Label: Hate
Model Predicted: Hate

D Dataset Train, Validation, and Test set Details

The full dataset size is 9843, with 6890 for training, 1476 for validation, and 1477 for testing.

E Adjusting Weights Logic

```
class_counts = np.bincount(labels_raw)
class_weights = 1. / class_counts
weights_tensor =
    torch.tensor(class_weights, dtype=torch.float)
     .to(device)
criterion =
      nn.CrossEntropyLoss(weight=weights_tensor)
```

Model	Metrics	Train/Test	Validation	Test
No Strategy	Macro-f1	0.717	0.578	0.608
	Macro-accuracy	0.757	0.621	0.632
	Macro-precision	0.733	0.602	0.628
	Macro-recall	0.731	0.563	0.594
With Oversampling	Macro-f1	0.984	0.252	0.244
	Macro-accuracy	0.985	0.411	0.392
	Macro-precision	0.983	0.243	0.231
	Macro-recall	0.986	0.265	0.260
With Undersampling	Macro-f1	0.809	0.236	0.231
	Macro-accuracy	0.806	0.329	0.319
	Macro-precision	0.789	0.259	0.251
	Macro-recall	0.856	0.219	0.215
With Adjusted Weights	Macro-f1	0.986	0.603	0.635
	Macro-accuracy	0.986	0.621	0.645
	Macro-precision	0.982	0.596	0.633
	Macro-recall	0.990	0.612	0.639

Table 6: Performance comparison of different training strategies for the Single Multiclass Classifier.

Metrics(Macro)	F1	Accuracy	Precision	Recall
Test(train split) set	0.717	0.757	0.733	0.731
Test(train split) set without emojis	0.984	0.985	0.984	0.985
(without emojis, with adjusted weights)	0.990	0.991	0.987	0.993
Validation File	0.578	0.621	0.602	0.563
Validation File without emojis	0.600	0.636	0.618	0.589
(without emojis, with adjusted weights)	0.589	0.608	0.585	0.594
Test File	0.608	0.632	0.628	0.594
Test File without emojis	0.622	0.640	0.635	0.613
(without emojis, with adjusted weights)	0.640	0.649	0.639	0.641

Table 7: Post Submission Performance metrics on evaluation and test datasets of Single Multiclass Classifier

Condition	Label	Macro-F1	Macro-Accuracy	Macro-Precision	Macro-Recall
With emojis replaced	Hope	0.73	0.77	0.72	0.74
	Hate	0.76	0.84	0.74	0.79
	Not Applicable	0.63	0.63	0.62	0.64
With no emojis	Hope	0.93	0.94	0.93	0.94
	Hate	0.95	0.97	0.96	0.94
	Not Applicable	0.92	0.91	0.91	0.91
Data split + emojis replaced	Hope	0.71	0.73	0.71	0.76
	Hate	0.78	0.84	0.76	0.82
	Not Applicable	0.63	0.62	0.62	0.67
Data split + no emojis	Hope	0.74	0.78	0.73	0.76
	Hate	0.78	0.86	0.78	0.77
	Not Applicable	0.65	0.66	0.65	0.65

Table 8: Performance of different setups for the Stacked Binary Ensemble Classifier on Test file.

MultiMinds at MAHED 2025: Multimodal and Multitask Approaches for Detecting Emotional, Hate, and Offensive Speech in Arabic Content

Riddhiman Swanan Debnath
Shahjalal University of Science and Technology, Sylhet

Abdul Wadud Shakib
Metropolitan University, Sylhet

Md Saiful Islam
Athabasca University, Canada

Abstract

This paper describes the MultiMinds team's participation in the MAHED 2025 shared task at ArabicNLP 2025, which targets the detection of hate speech, hope speech, and emotional expression in Arabic content. We addressed two subtasks. For the text-based subtask (Task 2), we experimented with multiple models, including Support Vector Machines with TF-IDF and AraBERT embeddings, XGBoost with fused AraBERT and XLM-RoBERTa embeddings optimized via Optuna, and a fine-tuned AraBERT model and GPT-5 (gpt-oss-20b). The fine-tuned AraBERT achieved the best performance with an F1 score of 0.68. For the multimodal subtask (Task 3), we proposed an architecture combining DistilBERT for text representation with a lightweight ELU-Net enhanced by a cross-attention mechanism, reaching 75% accuracy. Major challenges included dataset imbalance and noisy text, which we mitigated through preprocessing, class-weighted optimization, and feature fusion. Our results demonstrate the benefits of combining multiple embedding layers for text classification and leveraging lightweight multimodal architectures for robust hate speech detection in Arabic.

1 Introduction

Online media has become an important avenue for the consumption and distribution of information, and many people now rely on it as their primary source of news (Perrin, 2015). These have enabled individuals to share their views effortlessly through images and texts (multimodal and/or unimodal), reaching a broad and diverse audience (Fortuna and Nunes, 2018). With the rapid increase in media posts, manual detection of emotion, hate, and offensive (EHO) content becomes impractical. Consequently, there is a growing interest in developing automated methods for EHO detection.

MAHED 2025 (Zaghouani et al., 2025) is a shared task at ArabicNLP 2025 Co-located with EMNLP 2025, focusing on the detection of hate speech, hope speech, and emotional expression in Arabic content. Participants may choose to participate in one or more of the following three subtasks:(i) Text-based Hate and Hope Speech Classification, (ii) Emotion, Offensive, and Hate Detection (Multitask), and (iii) Multimodal Hateful Meme Detection. We, MultiMinds, participated in MAHED 2025, with particular interest in tasks (ii), (iii) and ranked 10th and 7th, respectively.

For Task (ii), three methods were tested for Arabic emotion, offensive, and hate-speech classification: Support Vector Machines (SVM) as a Baseline model with TF-IDF (best macro F1: 0.517); XGBoost with TF-IDF, AraBERT embeddings, and fused AraBERT and XLM-RoBERTa embeddings, which were optimized via Optuna (best F1: 0.57); and a deep learning approach fine-tuning AraBERT, which achieved the highest performance score. As the dataset was imbalanced and contained unnecessary information, the key challenge was to extract the correct information from the text. In our experiment for Task (iii), we used 1D-CNN model (Singh et al., 2021) as the Baseline model by extracting image and caption features by CLIP processor. Our enhanced ELU-Net architecture got the best results by incorporating a cross-attention mechanism to combine visual and textual features generated from the DistilBert (Sanh et al., 2019) tokenizer. Full Implementation here - Github. The main challenge of this task was that the classes were not equally distributed. Our key findings were as follows.

- Fusing multiple embedding layers from different textual models improves data representation.
- Using class weights enhances results.
- First-time use of a lightweight multimodal model to classify hateful and non-hateful memes.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 677–682
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

2.1 Emotion Detection

In recent years, research into developing state-of-the-art models for Arabic natural language processing tasks has gained momentum. Alswaidan and Menai (2020) proposed three models for emotion recognition in Arabic text. Abdullah et al. (2018) described their system - SEDAT, and showed substantial improvements in Spearman correlation scores over the baseline models. Alsmearat et al. (2015) explored the Gender Identification(GI) problem for Arabic text as a supervised learning problem and compared the Bag-Of-Words (BOW) approach with computing features related to sentiments and emotions. Biswas and Zaghouani (2025b) introduces a bilingual dataset comprising 23,456 entries for Arabic and 10,036 entries for English, annotated for emotions and hope speech, addressing the scarcity of multi-emotion (Emotion and hope) datasets. Al-Henaki et al. (2025) introduced MultiProSE, an open-source extension of the existing Arabic propaganda dataset, ArPro, with the addition of sentiment and emotion annotations for each text.

2.2 Offensive And Hate Speech Detection

While social media promotes free expression, it also fosters environments where hate speech spreads, making its detection a key research priority. Alsafari et al. (2020) built a reliable Arabic textual corpus by crawling data from Twitter. Mubarak et al. (2023) introduced a generic, language-independent method to collect a large percentage of offensive and hate tweets. Aldjanabi et al. (2021) developed a classification system for determining offensive and hate speech using a pre-trained Arabic language model. Biswas and Zaghouani (2025a) introduces multilabel hate speech dataset with offesnive content in the Arabic language. Zaghouani et al. (2024) analyzes 70,000 Arabic tweets, from which 15,965 tweets were selected and annotated, to identify hate speech patterns and train classification models.

2.3 MultiModal Hate Speech Detection

The usage of social media has enabled individuals to disseminate hateful messages through the use of memes. Chhabra and Vishwakarma (2023) highlighted handcrafted feature-based and deep learning-based algorithms by considering multimodal and multilingual inputs. Alam et al. (2024a) explored the intersection between propaganda and hate in memes using a multi-agent LLM-based approach. El-Sayed and Nasr (2024) described an approach to hateful meme classification for the Multimodal Hate Speech Shared Task at CASE 2024. Arya et al. (2024) introduced a novel approach by leveraging the CLIP model, fine-tuned through the incorporation of prompt engineering. Alam et al. (2024b) focused on developing an Arabic memes dataset with manual annotations of propagandistic content. AlDahoul and Zaki (2025) explores the potential of large language models to effectively identify hope, hate speech, offensive language, and emotional expressions. Kmainasi et al. (2025) introduced MemeIntel, an explanation-enhanced dataset for propaganda memes in Arabic and hateful memes in English. However, multimodal hate speech detection lacks the use of lightweight architectures.

3 System Overview

Before tackling Task 2, we observed that the dataset (Zaghouani et al., 2024), (Biswas and Zaghouani, 2025b), (Biswas and Zaghouani, 2025a) was both imbalanced and noisy. To address the noise, we performed text cleaning and preprocessing, converting the text into TF–IDF features and tokenizing it using the AraBERT tokenizer. We then fused the embedding layers of XLM-RoBERTa (Conneau et al., 2019) and AraBERT (Antoun et al., 2020). Furthermore, to mitigate the impact of class imbalance, we incorporated class distribution-based weighting. For preprocessing, we compiled Arabic and English punctuation, removed Arabic diacritics via regex [1], eliminated repeated characters, English words, and numbers, and collapsed multiple spaces into one for clean tokenization. Arabic characters were standardized to reduce variations, ensuring a consistent representation of letters that look or sound similar; for example, different forms of Alif (إ، أ، آ، ا) were replaced with the standard form ا (U+0627).

For feature extraction, we used TF-IDF (Jalilifard et al., 2021) with the top 5,000 terms (unigrams and bigrams). AraBERT and XLM-RoBERTa embeddings were integrated with a 128-token limit, applying padding and truncation, and extracting the [CLS] token from the final hidden state. To fine-tune GPT-5 (Daniel Han and team, 2023), we employ LoRA adapters within the PEFT

[1] https://docs.python.org/3/howto/regex.html

framework, incorporating a curated set of few-shot examples.

For Task 3, we employed the CLIP via Radford et al. (2021) processor for feature extraction, utilizing the ViT-B/32 [2] transformer architecture as the image encoder and a masked self-attention transformer as the text encoder. The extracted multimodal features were fed into a Support Vector Machine for classification; it failed to identify hateful memes accurately. The main challenge was dataset (Alam et al., 2024a), (Alam et al., 2024b) imbalance, which could be mitigated by collecting more hateful memes for a balanced distribution. Additionally, as non-Arabic speakers, understanding the language and cultural context was difficult, so we relied on a CNN-based neural network for better performance.

To achieve our objective of developing a lightweight model, we employed the ELUNet architecture via Deng et al. (2022). Since all captions in the dataset are in the Arabic language, textual features were extracted using the DistilBERT tokenizer via Devlin et al. (2018). In the case of preprocessing and cleaning, the same procedure as Task 2 was followed. Another challenge we faced was that the tokenizers' lengths were not equal for all memes, as they hold different sizes of text. So we fixed the tokenizer size to 256. If the tokenizer length is smaller than the value, the previous value will repeat; otherwise larger size tokenizer will be shrunk using the PCA algorithm (Drikvandi and Lawal, 2023). The corresponding images were processed through the encoder component of the ELUNet architecture. Inspired by Li et al. (2024), a cross-attention mechanism was then applied, integrating the encoded image features from the encoder with the textual embeddings generated by the tokenizer, positioned at the intermediate layers of ELUNet. The cross-attention outputs were subsequently passed through the decoder component of ELUNet. The proposed model (Figure 1) produces two outputs.

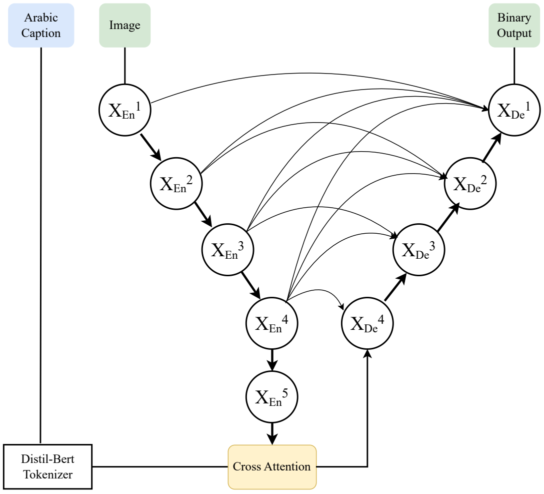


Figure 1: The architecture of Attention-based ELUNet

4 Experimental Setup

4.1 Emotion, Offensive Language, and Hate Detection

The whole dataset was split into Train(70%), Test(15%), and Validation(15%) via stratified sampling across emotion, hate, and offensive tasks, with exception for GPT-5 (80-10-10). Table 1 provides a brief overview of various emotions in the dataset, including its size and distribution of various emotions, as well as there are offensive (yes - 1744, no - 4216) and hate (yes - 303, no - 1441). Table 2 reveals the Task 2 dataset contains the most non-Arabic characters (see Figure 2).

Name	Amount
Anger	1551
Disgust	777
Neutral	661
Love	593
Joy	533
Anticipation	491
Optimism	419
Sadness	335
Confidence	210
Pessimism	194
Surprise	143
Fear	53

Table 1: Emotion Proportions in Training Data – Task 2

We used Optuna with a class-weighted objective to optimize XGBoost hyperparameters for the highest macro F1-score. We incorporated a deep learning approach using AraBERTv2 [3] for multitask classification across emotion, offensive language, and hate speech tasks. Three task-specific linear layers mapped the 768-dimensional hidden representation to class logits, with dropout applied to improve generalization. For fine-tuning GPT-5, we

[2] https://huggingface.co/openai/clip-vit-base-patch32

[3] https://huggingface.co/aubmindlab/bert-base-arabertv2

Name	Non-Arabic Chars Count
Train (Task 2)	157138
Test	32968
Validation	32075
Train (Task 3)	4737
Test	1340
Validation	1310

Table 2: Non-Arabic Characters in Tasks 2 & 3

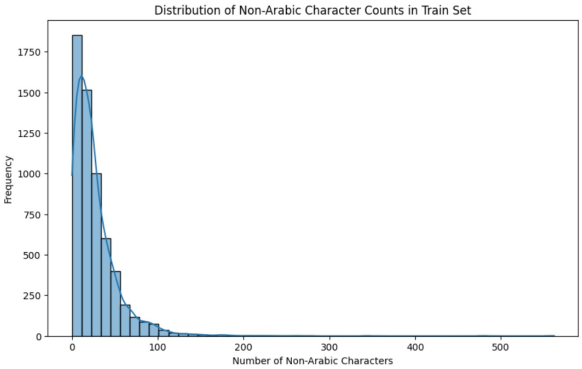


Figure 2: Non-Arabic Character Distribution – Train Set (Task 2)

configured the rank, selected specific transformer layers, and applied an appropriate scaling factor, while enabling gradient checkpointing to optimize memory usage. Furthermore, no bias parameters were introduced to ensure that the fine-tuning process remained lightweight.

	Emo	Offn	Hate
learning rate	0.0060	0.0011	0.0037
max depth	10	7	10
num. estimator	50	282	182
subsample	0.9453	0.8231	0.7524
colsample_bytree	0.7366	0.6489	0.8440
scale_pos_weight	**x**	2.4179	0.0535

Table 3: Best parameter value from trial run

4.2 Multimodal Hate Speech Detection in Memes

Table 4 presents the distribution of hateful content in training, development, and test sets. We processed each meme (text + image) using CLIP to create joint features. Text was tokenized and images scaled to RGB via CLIPProcessor, producing tensors for both modalities. Features were concatenated and fed to a 1D-CNN. Then we evaluated our enhaced ELUNet model with AraBert, DistilBert tokenizers. Our best model, ELUNet with the DistilBert tokenizer gave the accuracy of 75%. In our experiment, we chose batch size 16, epoch 5, and learning rate 10^{-3}. This model was trained in Google Colab and consumed 6.2 GB of GPU.

Name	Hate	Not Hate
Train	213	1930
Dev	31	281
Test	154	452
Total	398	2663

Table 4: Dataset Size – Task 3 (Initial)

5 Results

Table 5 summarizes our model's performance on the task 2 dataset. The results indicate that applying class weights improves performance based on the average F1 score, while incorporating deep learning approaches yields even higher results. For instance, in our experiments with AraBERT, using a batch size of 8, 5 epochs, a dropout rate of 0.3, and a learning rate of 10^{-5} with the exception (10^{-4}) for Gpt-5, we achieved an F1 score of 0.67. Reducing dropout to 0.1, while doubling both batch size and epochs, increased the score to 0.68, matching the performance of DistilBERT. However, with respect to accuracy, GPT-5 and AraBERT achieved comparable performance on the offensive and hate detection tasks, while exhibiting notable differences in the emotion classification task.

App.	Model	Emo	Offn	Hate	Avg
Without Weight	XGB	0.172	0.416	0.344	0.312
	XGB-AraBERT	0.241	0.712	0.541	0.484
	XGB-AraBERT+XLMRoBERTa	0.244	0.414	0.500	0.384
	SVM(Baseline)	0.284	0.702	0.564	**0.513**
With Weight	XGB	0.212	0.712	0.400	0.393
	XGB-AraBERT+XLMRoBERTa	0.264	0.723	0.500	0.493
	XGB-AraBERT+XLMRoBERTa Trial	0.324	0.775	0.624	**0.574**
DL	AraBERT	0.267	0.834	0.954	**0.684**
	DistilBERT	**0.373**	0.774	0.924	**0.683**
	Gpt-oss-20b (PC)	0.014	0.412	0.483	0.300

Table 5: Performance of the models on the Task 2 dataset. Here, PC, Emo, Offn, Hate, and Avg denote the post-competition, emotion, offensive, hate, and average macro F1 scores, respectively.

The model performances in Task 3 are described in Table 6. For adding class weight, the result has been improved. Finally, we get an accuracy of 75%. For each testing section test dataset was utilized. Despite fixing the epoch to 20, the best-fitting model took only 5 epochs by using the early stopping concept.

Model	Acc	MacroAvg-f1	Hateful(f1)	Non-Hateful(f1)
1D-CNN(Baseline)	0.745	0.431	0	0.851
ELUNet-DistilBert	0.746	0.421	0	0.852
ELUNet-AraBert	0.744	0.422	0	0.853
ELUNet-AraBert (WW)	0.746	0.372	0	0.855
ELUNet-DistilBert(WW)	0.754	0.500	0.165	0.858

Table 6: Performance of the models on the Task 3 dataset. Here, WW represents 'with weight'.

6 limitations

Both subtasks (Task 2: Emotion, Offensive, and Hate Detection; Task 3: Multimodal Hateful Meme Detection) suffered from severe class imbalance. This led to biased models, poor performance on minority classes, and necessitated mitigations such as class weighting, which still did not fully resolve the issue. Fine-tuning was limited (e.g., 5 epochs with early stopping, a fixed tokenizer length of 256, and PCA for shrinkage), which may have led to underfitting. GPT-5 experiments were constrained by few-shot examples and memory optimizations (e.g., LoRA adapters), resulting in lower emotion detection scores (F1=0.014).

7 Conclusion

Our participation in MAHED 2025 highlights the effectiveness of advanced NLP and multimodal methods for detecting hate speech, hope speech, and emotions in Arabic. For Task 2, our fine-tuned AraBERT scored 0.68 macro F1, surpassing SVM and XGBoost baselines through class-weighted optimization and fused embeddings to address imbalance and noise. For Task 3, our lightweight ELU-Net, cross-attention with tokenizer generated from DistilBert, achieved 75 % accuracy on hateful meme classification despite imbalance. Challenges included limited Arabic meme data, non-Arabic characters, and noisy text affecting preprocessing and features. Future work will explore data augmentation, advanced multimodal fusion, and improved preprocessing and fine-tuning to boost robustness and generalization.

References

Malak Abdullah, Mirsad Hadzikadicy, and Samira Shaikhz. 2018. Sedat: sentiment and emotion detection in arabic text using cnn-lstm deep learning. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 835–840. IEEE.

Lubna Al-Henaki, Hend Al-Khalifa, Abdulmalik Al-Salman, Hajar Alqubayshi, Hind Al-Twailay, Gheeda Alghamdi, and Hawra Aljasim. 2025. Multiprose: A multi-label arabic dataset for propaganda, sentiment, and emotion detection. In *International Conference on Applications of Natural Language to Information Systems*, pages 156–172. Springer.

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmad, Md. Arid Hasan, and Maram Hasanain. 2024b. ArMeme: Propagandistic content in Arabic memes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21071–21090, Miami, Florida, USA. Association for Computational Linguistics.

Nouar AlDahoul and Yasir Zaki. 2025. Detecting hope, hate, and emotion in arabic textual speech and multimodal memes using large language models. *arXiv preprint arXiv:2508.15810*.

Wassen Aldjanabi, Abdelghani Dahou, Mohammed AA Al-Qaness, Mohamed Abd Elaziz, Ahmed Mohamed Helmi, and Robertas Damaševičius. 2021. Arabic offensive and hate speech detection using a cross-corpora multi-task learning model. In *Informatics*, volume 8, page 69. MDPI.

Safa Alsafari, Samira Sadaoui, and Malek Mouhoub. 2020. Hate and offensive speech detection on arabic social media. *Online Social Networks and Media*, 19:100096.

Kholoud Alsmearat, Mohammed Shehab, Mahmoud Al-Ayyoub, Riyad Al-Shalabi, and Ghassan Kanaan. 2015. Emotion analysis of arabic articles and its impact on identifying the author's gender. In *2015 IEEE /ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–6. IEEE.

Nourah Alswaidan and Mohamed El Bachir Menai. 2020. Hybrid feature model for emotion recognition in arabic text. *IEEE Access*, 8:37843–37854.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Greeshma Arya, Mohammad Kamrul Hasan, Ashish Bagwari, Nurhizam Safie, Shayla Islam, Fatima Rayan Awad Ahmed, Aaishani De, Muhammad Attique Khan, and Taher M Ghazal. 2024. Multimodal hate speech detection in memes using contrastive language-image pre-training. *IEEE Access*, 12:22359–22375.

Md. Rafiul Biswas and Wajdi Zaghouani. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *CoRR*, abs/2505.11969.

Md. Rafiul Biswas and Wajdi Zaghouani. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *CoRR*, abs/2505.11959.

Anusha Chhabra and Dinesh Kumar Vishwakarma. 2023. A literature survey on multimodal and multilingual automatic hate speech identification. *Multimedia Systems*, 29(3):1203–1230.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Yunjiao Deng, Yulei Hou, Jiangtao Yan, and Daxing Zeng. 2022. Elu-net: An efficient and lightweight u-net for medical image segmentation. *IEEE Access*, 10:35932–35941.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Reza Drikvandi and Olamide Lawal. 2023. Sparse principal component analysis for natural language processing. *Annals of data science*, 10(1):25–41.

Ahmed El-Sayed and Omar Nasr. 2024. Aast-nlp at multimodal hate speech event detection 2024: A multimodal approach for classification of text-embedded images based on clip and bert-based models. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 139–144.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *Acm Computing Surveys (Csur)*, 51(4):1–30.

Amir Jalilifard, Vinicius Fernandes Caridá, Alex Fernandes Mansano, Rogers S Cristo, and Felipe Penhorate Carvalho da Fonseca. 2021. Semantic sensitive tf-idf to determine word relevance in documents. In *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 2*, pages 327–337. Springer.

Mohamed Bayan Kmainasi, Abul Hasnat, Md Arid Hasan, Ali Ezzat Shahroor, and Firoj Alam. 2025. Memeintel: Explainable detection of propagandistic and hateful memes. *arXiv preprint arXiv:2502.16612*.

Hongchan Li, Yantong Lu, and Haodong Zhu. 2024. Multi-modal sentiment analysis based on image and text fusion based on cross-attention mechanism. *Electronics*, 13(11):2069.

Hamdy Mubarak, Sabit Hassan, and Shammur Absar Chowdhury. 2023. Emojis as anchors to detect arabic offensive language and hate speech. *Natural Language Engineering*, 29(6):1436–1457.

Andrew Perrin. 2015. Social media usage. *Pew research center*, 125:52–68.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Kuljeet Singh, Amit Mahajan, and Vibhakar Mansotra. 2021. 1d-cnn based model for classification and analysis of network attacks. *International Journal of Advanced Computer Science and Applications*, 12(11):604–613.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgio Mikros, and Firoj Alam. 2025. Overview on mahed 2025 shared task: Multimodal detection of hope and hate emotions in arabic content. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

joy_2004114 at MAHED Shared Task : Filtering Hate Speech from Memes using A Multimodal Fusion-based Approach

Joy Das, Alamgir Hossain and Mohammed Moshiul Hoque
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology
{u2004114, 23mcse701}@student.cuet.ac.bd
moshiul_240@cuet.ac.bd

Abstract

Social media platforms have become major spaces for sharing opinions, humor, and information through memes that blend images with text. While many memes are harmless, some promote hate speech against individuals or communities based on cultural, religious, gender, or national identity. Detecting such content in Arabic is particularly challenging due to linguistic complexity, cultural context, and limited annotated data. In this study, we present an effective approach for detecting hateful content in Arabic memes using the **QCRI Prop2Hate-Meme** dataset, which contains image–text pairs labeled for hatefulness. We experimented with several multimodal configurations, and the best performance was achieved using a combination of InceptionNet for visual features and multilingual BERT for text. These representations were fused after applying normalization and augmentation to enhance robustness. Our **InceptionNet** with **mBERT** configuration achieved a macro F1-score of 63 percent and secured the sixth position on the official CodaBench leaderboard. These findings highlight the strength of our multimodal model and support its potential for detecting harmful Arabic content in low-resource settings.

1 Introduction

With the exponential growth of social media platforms, memes have evolved into a dominant means of communication, often blending visual and textual elements to express humor, satire, or commentary. However, this same format has been increasingly exploited to propagate hateful narratives targeting individuals or communities based on attributes such as culture, religion, sex or nationality (Kiela et al., 2021; Pramanick et al., 2021; Sharma et al., 2020a). Unlike conventional text based hate speech, hateful memes present a unique detection challenge since the offensive intent may only emerge when text and image are interpreted together (Das et al., 2020; Zhao et al., 2023), making unimodal approaches insufficient.

In Arabic speaking contexts, the task is further complicated by several factors. First, there is a scarcity of large scale, high quality annotated datasets for multimodal hate speech detection (Zaghouani et al., 2025) . Second, most state-of-the-art detection models have been trained primarily on English datasets, limiting their transferability due to linguistic, cultural, and script specific nuances . Text only models risk overlooking visual sarcasm or symbolism, while image only systems may fail to capture hateful meaning embedded in overlaid text, leading to both false positives and false negatives.

The NeurIPS "Hateful Memes" Challenge (Kiela et al., 2021) highlighted how benign confounders, individually innocuous text and images that form hateful meaning only when combined, require models to perform genuine multimodal reasoning. Large scale vision language transformers such as **UNITER** (Chen et al., 2020), **ViLT** (Kim et al., 2021), and **CLIP** (Arya et al., 2024; Radford et al., 2021) have achieved strong results in high-resource settings but their reliance on vast amounts of paired data and high computational cost renders them impractical for low resource languages like Arabic. While dual-encoder fusion strategies (Ahsan et al., 2024; Hossain et al., 2022; Lippe et al., 2020; Zhou et al., 2021) have shown promising performances in other languages, systematic evaluations for Arabic meme moderation remain scarce.

In order to overcome these challenges, we proposed a lightweight dual-encoder multimodal framework that combined a fine-tuned **Inception-ResNetV2** image encoder (Szegedy et al., 2016) with a **multilingual BERT (mBERT)** text encoder (Pires et al., 2019). Visual features were extracted from resized meme images, while textual features were derived from OCR-extracted Arabic text after normalization (Kaundilya et al., 2019; Hossein-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 683–691
November 8-9, 2025 ©2025 Association for Computational Linguistics

mardi et al., 2015). These representations were concatenated and passed through a compact multilayer perceptron for binary classification, following prior dual-encoder fusion strategies in multimodal hate speech detection (Pramanick et al., 2021; Ahsan et al., 2024).The design maintained a trade-off between accuracy and efficiency, making it practical for use in environments with limited computational resources.

The key contributions of this study are :

- We developed a lightweight multimodal architecture for Arabic hateful meme detection by integrating InceptionResNetV2 with mBERT.
- We conducted comparative experiments across multiple model combinations and found that InceptionNet + mBERT achieved the best macro F1.
- We designed a preprocessing pipeline with OCR-based text extraction, normalization, and image augmentation to improve robustness.

2 Background & Related Work

2.1 Task Definition

We participated in **Subtask 3: Multimodal Hateful Meme Detection**, which is part of **Shared Task 4 (MAHED 2025: Multimodal Detection of Hope and Hate Emotions in Arabic Content)**, organized under **Track 1: Speech and Multimodal Processing** at the **ArabicNLP 2025** workshop. We used the QCRI/Prop2Hate-Meme[1](Alam et al., 2024), which was released for this shared task. This subtask focuses on classifying Arabic memes that contain both images and text as either *hateful* or *non-hateful*. Each sample contains:

1. **Image**: Visual content, symbols, or scenes conveying context or sentiment.
2. **Embedded Arabic text**: Extracted text from the image, providing essential linguistic context.

A meme is classified as hateful if it explicitly or implicitly promotes hostility, discrimination, or stereotypes toward a targeted group. Non-hateful memes lack such harmful content, even when expressing strong opinions or satire.

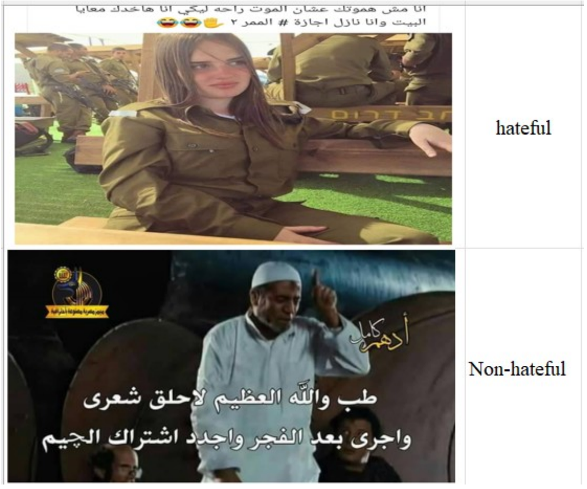

انا مش هموتك عشان الموت راحه ليكي انا هاخدك معايا
البيت وانا نازل اجازة # الممر ٢
hateful
أدهم كامل
طب والله العظيم لاحلق شعرى
واجرى بعد الفجر واجدد اشتراك الجيم
Non-hateful

Figure 1: Examples of hateful and non-hateful memes.

Formally, the problem is modeled as a binary classification task:

$$f(image, text) \rightarrow \{hateful, non - hateful\}$$

The key challenge lies in effectively capturing the interplay between visual and textual information, as hateful intent often arises from their combined interpretation rather than either modality alone.

2.2 Related Work

Detecting hateful memes inherently requires joint vision–language reasoning. The Hateful Memes benchmark introduced by Kiela (Kiela et al., 2021) incorporated benign confounders, text and images that appear harmless in isolation but convey hateful meaning when combined, to prevent models from exploiting unimodal shortcuts. Their multimodal baselines, such as late-fusion BERT(Devlin et al., 2019) + ResNet(He et al., 2015), achieved around 65% accuracy, whereas unimodal baselines rarely exceeded 60%. The follow-up competition report showed that improved cross-modal grounding, use of auxiliary supervision and more effective fusion strategies enabled some teams to surpass 70% accuracy.

Subsequent works explored architectural variations. Zhou (Kiela et al., 2021) integrated auxiliary image and text matching tasks with a BERT + ResNet pipeline, reporting macro F1 gains of approximately 3–4 percentage points over vanilla fusion. Lippe (Lippe et al., 2020) employed contrastive learning to better align modalities, improving robustness to adversarial confounders and achieving competitive leaderboard rankings.

[1] https://huggingface.co/datasets/QCRI/Prop2Hate-Meme

Large-scale vision–language transformers such as ViLBERT (Lu et al., 2019), VisualBERT (Li et al., 2019), UNITER, ViLT, and CLIP have demonstrated strong modality alignment across a range of multimodal tasks. These models were not originally trained on the Hateful Memes dataset, but some have been later fine-tuned for it and achieved over 70 percent accuracy. However, their reliance on large paired datasets and substantial computational resources limits their suitability for low-resource language settings.

In the related ArAIEval 2024 propaganda meme classification task (Hasanain et al., 2024), the top multimodal system (AlexUNLP-MZ)(Zaytoon et al., 2024) reached a macro F1 of 0.8051. These results demonstrate the benefit of combining visual and textual features for Arabic meme moderation. Despite this progress, lightweight dual-encoder fusion commonly explored for English and multilingual settings which remains under investigated for Arabic multimodal content moderation.

Cross-lingual studies further demonstrate the viability of compact fusion strategies. Datasets such as Memotion(Sharma et al., 2020b) and MUTE (Hossain et al., 2022) have enabled systematic benchmarking, while dual-encoder pipelines (Ahsan et al., 2024) and CLIP-based transfer learning can perform competitively in low-resource contexts, with reported results ranging from 60–68% depending on modality balance and data quality.

Building on these findings, our work adopts a lightweight vision backbone combined with multilingual BERT to balance accuracy, OCR robustness, and deployability in resource constrained settings for Arabic hateful meme detection.

3 Dataset

We use the QCRI/Prop2Hate-Meme corpus released for the Arabic multimodal meme shared task. The dataset pairs each meme image with aligned Arabic text and provides both coarse (hateful vs. non-hateful) and fine-grained hatefulness labels, while preserving related propaganda annotations. Appendix A shows the dataset's statistics, including important measures and how the data is spread out.

4 Methodology

Our approach combines text and image information to detect hateful content in Arabic memes. The workflow begins with preprocessing of both text and image inputs, followed by feature extraction using pre-trained models, and finally a fusion step that integrates the two modalities for classification. An outline of the complete pipeline is presented in Figure 2, which illustrates how data flows through preprocessing, feature extraction, and multimodal fusion before reaching the classifier.

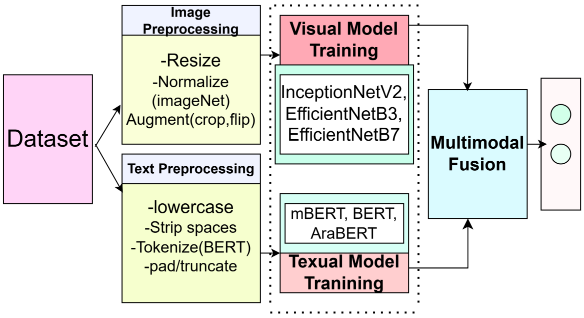


Figure 2: An overview of the methodology for our proposed system

4.1 Data Preprocessing

The experiments employed the publicly available QCRI/Prop2Hate-Meme dataset, hosted on the Hugging Face repository. The dataset contains multimodal entries, each consisting of a meme image and associated text, annotated for binary hate speech classification ($hate\_label \in \{0, 1\}$). The dataset was supplied in `Parquet` format and pre-split into training, validation, and test sets.

Text preprocessing: For each meme text, we first measured its length in characters and recorded both the maximum and median values. Based on these lengths, the data was divided into three intervals to analyze distribution patterns. To reduce noise, extremely long non-hateful samples (those exceeding 62 characters) were excluded, while all hateful samples were kept to preserve minority class information. Texts were converted to lowercase, unnecessary spaces were removed, and the content was tokenized using the BERT tokenizer. Finally, each sequence was padded to a maximum length of 128 tokens.

Image preprocessing: All meme images were decoded from their byte format using the PIL library and then resized to a resolution of 128 × 128 pixels to maintain consistency. The resized images were converted into NumPy arrays, and their pixel values were normalized to fall within the range [0, 1]. Finally, the processed images were stored as stacked NumPy arrays, allowing efficient loading. All meme images were first decoded from their byte format using the PIL library and then resized to a fixed resolution of 128 × 128 pixels to maintain consistency across samples. The resized images were converted into NumPy arrays, and their pixel values were normalized to fall within the range [0, 1] for stable model training. Finally, the processed images were stored as stacked NumPy arrays, allowing efficient loading and batch processing during training.

4.2 Feature Extraction

Text was encoded with the bert-base-multilingual-cased transformer. After tokenization, batches of 32 samples were forwarded through the model, and the [CLS] representation from the hidden layer was taken as the sentence-level embedding. The resulting vectors were stored as NumPy arrays of shape $(N, 768)$ for downstream use. Text was encoded with the bert-base-multilingual-cased transformer. After tokenization (max length 128 with padding/truncation), batches of 32 samples were forwarded through the model, and the [CLS] representation from the final hidden layer was taken as the sentence-level embedding. The resulting vectors were stored as NumPy arrays of shape $(N, 768)$ for downstream use.

Images were processed with a pre-trained InceptionResNetV2 backbone (ImageNet weights) with the classification head removed (include_top=False). Preprocessed inputs of size 128×128 were passed through the network, and a Global Average Pooling layer yielded a 1536-dimensional descriptor per image. This descriptor was flattened to obtain a fixed-length visual feature vector. The text and image embeddings produced here serve as inputs to the subsequent multimodal fusion module.

4.3 Baselines

Different unimodal models (image only and text only) and multimodal models (combining image and text) were analyzed and fused using an **early fusion** strategy with appropriate hyperparameter tuning.

4.3.1 Unimodal Baselines

To extract textual features, we utilized AraBERT, mBERT, and BERT. For visual features, we experimented with InceptionResNetV2, EfficientNetB3, and EfficientNetB7. The effectiveness of these models was assessed before integration into the multimodal framework. Various deep learning models were employed to establish unimodal baselines. For textual feature extraction, we utilized AraBERT(Antoun et al., 2020), mBERT, and BERT. For visual features, we experimented with InceptionResNetV2, EfficientNetB3, and EfficientNetB7(Tan and Le, 2020). These models were individually trained and evaluated to assess their effectiveness before integration into the multimodal framework. Appendix B.1 outlines the hyperparameters configured for both the textual and visual unimodal models.

4.3.2 Multimodal Baselines

We adopt an early fusion strategy where the 1536-D image vector and the 768-D text embedding are concatenated to form a 2304-D joint representation. The fused vector is then passed through fully connected layers with ReLU activations (1024 → 512 → 256 → 128), with dropout (0.5) applied after the 512- and 128-unit layers to mitigate overfitting. Finally, a dense layer with two units and a softmax activation outputs the class probabilities for hateful vs. non-hateful.

We evaluated several multimodal combination models, including InceptionNet + mBERT, InceptionNet + BERT, InceptionNet + AraBERT, EfficientNetB3 + mBERT, and EfficientNetB7 + mBERT. The hyperparameters used in this work include learning rate, number of epochs, batch size, dropout rate, optimizer and activation function, as summarized in the appendixB.2.

5 Results Analysis

The proposed multimodal fusion approach was evaluated on the official test split of the QCRI/Prop2Hate-Meme dataset as part of the shared task hosted on CodaBench . Among the tested configurations, the best-performing setup, combining InceptionNet (Szegedy et al., 2016) with mBERT (Devlin et al., 2019), achieved a macro F1-score of 69%, outperforming several strong vision–language baselines commonly used in hate speech detection tasks. (Kiela et al., 2021; Gomez et al., 2020)

Table 1 provides a comparative evaluation of textual and visual models. Within the text-only approaches, AraBERT and mBERT demonstrated the highest performance, both reaching an F1-score of

[1] Source code available at: GitHub Repository

Approaches	Classifiers	F1	P	R	G
Textual only	**AraBERT**	**0.63**	0.61	0.69	0.64
	BERT-base-uncased	0.55	0.57	0.67	0.61
	mBERT	**0.63**	0.61	0.69	0.64
Visual only	EfficientNetB3	0.45	0.54	0.61	0.57
	EfficientNetB7	0.48	0.58	0.62	0.60
	InceptionNetV2	**0.58**	0.57	0.60	0.58

Table 1: Result comparison on validation data of unimodal models, where F1, P, R, and G represent F1-score, precision, recall, and the geometric mean of precision and recall, respectively

0.63. In the visual-only category, InceptionNetV2 achieved the best result with an F1-score of 0.58.

Classifiers	F1	P	R	G	F_F1
InceptionNet + AraBERT	0.59	0.56	0.70	0.62	0.59
InceptionNet + BERT-base-uncased	0.67	0.75	0.61	0.68	0.60
EfficientNetB3 + mBERT	0.56	0.60	0.53	0.55	0.56
EfficientNetB7 + mBERT	0.67	0.70	0.64	0.67	0.60
InceptionNet + mBERT(Proposed)	**0.69**	0.66	0.76	0.71	**0.63**

Table 2: Result comparison on validation data of multimodal models, where F1, P, R, G, and F_F1 denote F1-score, precision, recall, geometric mean of precision and recall, and official F1-score, respectively

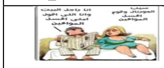
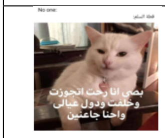
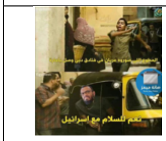
In multimodal comparative evaluations, as shown in Table 2, InceptionNet + AraBERT obtained 59%, InceptionNet + BERT-base-uncased (Devlin et al., 2019) reached 67%, EfficientNetB3 + mBERT scored 56%, and EfficientNetB7 + mBERT achieved 67% macro F-1 score. InceptionNet + mBERT achieved the highest score of 69% among all tested architectures, demonstrating its effectiveness in jointly leveraging visual and textual cues for hateful meme detection in Arabic content. The official shared task result was 63%, securing 6th place on the leaderboard.

5.1 Error Analysis

We present representative examples of both correctly and incorrectly classified hateful and non-hateful memes. Misclassifications often result from subtle visual cues, implicit expressions, or cases where the hateful intent is weak or context-dependent. The selected instances are summarized in Table 3, which lists the input meme, its true label, and the model's prediction.

Detailed error analysis is provided in the appendix C.

Table 3: Examples of predicted outputs

Input Meme	True Label	Predicted Label
 انا راجل البيت وانا اللي اقول امتى اغسل المواعين سيبي الجورنال وقوم اغسل المواعين	Hateful	Non-Hateful
 No one: بصي انا رحت اتجوزت وخلفت ودول عيالى واحنا جاعنين	Non-Hateful	Non-Hateful
	Non-Hateful	Hateful
	Hateful	Hateful

6 Conclusion and Future Work

This study presented a multimodal fusion approach for detecting hateful content in Arabic memes by combining visual and textual information. Using the QCRI/Prop2Hate-Meme dataset, our best-performing configuration is InceptionNet with mBERT which achieved a **macro F1-score of 63%** in the official Codabench shared task evaluation. Results show that multimodal integration significantly outperforms unimodal models, especially where meaning depends on text-image interaction. Future work includes exploring advanced fusion methods, Vision-Language Models (VLMs), advance data augmentation to mitigate class imbalance, leveraging external context for subtle cues, and extending to multilingual scenarios.

Limitations

This work is limited to deep learning and transformer models, excluding traditional machine learning comparisons. The imbalanced dataset without advanced augmentation may have led to biased predictions. While multimodal fusion improved results, it also increased overfitting risks and computational costs.

References

Shawly Ahsan, Eftekhar Hossain, Omar Sharif, Avishek Das, Mohammed Moshiul Hoque, and M Dewan. 2024. A multimodal framework to detect target aware aggression in memes. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2487–2500.

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

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.

Greeshma Arya, Mohammad Kamrul Hasan, Ashish Bagwari, Nurhizam Safie, Shayla Islam, Fatima Rayan Awad Ahmed, Aaishani De, Muhammad Attique Khan, and Taher M Ghazal. 2024. Multimodal hate speech detection in memes using contrastive language-image pre-training. *IEEe Access*, 12:22359–22375.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning.

Abhishek Das, Japsimar Singh Wahi, and Siyao Li. 2020. Detecting hate speech in multi-modal memes.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2020. Exploring hate speech detection in multimodal publications. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1459–1467.

Maram Hasanain, Md. Arid Hasan, Fatema Ahmed, Reem Suwaileh, Md. Rafiul Biswas, Wajdi Zaghouani, and Firoj Alam. 2024. Araieval shared task: Propagandistic techniques detection in unimodal and multimodal arabic content.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. Mute: A multimodal dataset for detecting hateful memes. In *Proceedings of the 2nd conference of the asia-pacific chapter of the association for computational linguistics and the 12th international joint conference on natural language processing: student research workshop*, pages 32–39.

Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2015. Analyzing labeled cyberbullying incidents on the instagram social network. In *International conference on social informatics*, pages 49–66. Springer.

Chandni Kaundilya, Diksha Chawla, and Yatin Chopra. 2019. Automated text extraction from images using ocr system. In *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 145–150. IEEE.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2021. The hateful memes challenge: Detecting hate speech in multimodal memes.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language.

Phillip Lippe, Nithin Holla, Shantanu Chandra, Santhosh Rajamanickam, Georgios Antoniou, Ekaterina Shutova, and Helen Yannakoudakis. 2020. A multimodal framework for the detection of hateful memes.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert?

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. MOMENTA: A multimodal framework for detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020a. SemEval-2020 task 8: Memotion analysis- the visuo-lingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online). International Committee for Computational Linguistics.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020b. SemEval-2020 task 8: Memotion analysis- the visuo-lingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online). International Committee for Computational Linguistics.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning.

Mingxing Tan and Quoc V. Le. 2020. Efficientnet: Rethinking model scaling for convolutional neural networks.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgio Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Mohamed Zaytoon, Nagwa El-Makky, and Marwan Torki. 2024. AlexUNLP-MZ at ArAIEval shared task: Contrastive learning, LLM features extraction and multi-objective optimization for Arabic multimodal meme propaganda detection. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 512–517, Bangkok, Thailand. Association for Computational Linguistics.

Bryan Zhao, Andrew Zhang, Blake Watson, Gillian Kearney, and Isaac Dale. 2023. A review of vision-language models and their performance on the hateful memes challenge.

Yi Zhou, Zhenhao Chen, and Huiyuan Yang. 2021. Multimodal learning for hateful memes detection. In *2021 IEEE International conference on multimedia & expo workshops (ICMEW)*, pages 1–6. IEEE.

A Dataset Statistics

We follow the official train, development, and test splits without modification. The memes cover common Arabic social topics, including politics, public figures, religion, security, and social issues, making the dataset representative for real-world vision–language tasks. The training set contains 2,143 samples (1,930 non-hateful, 213 hateful), the development set has 312 samples (281 non-hateful, 31 hateful), and the test set includes 603 samples (452 non-hateful, 151 hateful).

Hate label	Train	Dev	Test	W_T	UW_T
Non_hateful	1930	281	452	36123	18059
Hateful	213	31	151	6484	4770
Total	2143	312	603	42607	22829

Table A.1: Class distribution of training, development, and test sets. W_T denotes total words and UW_T denotes unique words for each class.

Table A.1 shows that the dataset suffers from a notable class imbalance, with non-hateful samples dominating across all splits. This imbalance poses challenges for training reliable classifiers and highlights the need for robust evaluation strategies.

B Hyperparameter Setting

B.1 Unimodal Hyperparameters

Table B.1 summarizes the hyperparameters used for the textual models (AraBERT, mBERT, and BERT). The parameters include the learning rate, number of epochs, batch size, dropout rate, optimizer, and activation function. These values were selected after systematic tuning to achieve stable and consistent performance across the text-only experiments.

Hyperparameter	AraBERT	mBERT	BERT
Dropout rate	0.5	0.5	0.5
Optimizer	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001
Epochs	16	16	20
Batch size	32	32	32

Table B.1: Hyperparameters used for training the textual models

Hyperparameter	InceptionNet	EfficientNetB3	EfficientNetB7
Dropout rate	0.5	0.5	0.5
Optimizer	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001
Epochs	16	16	20
Batch size	32	32	32

Table B.2: Hyperparameters used for training the visual models

Table B.2 presents the hyperparameters adopted for the visual feature extraction networks (InceptionResNetV2, EfficientNetB3, and EfficientNetB7). Similar to the textual models, we optimized learning rate, epochs, batch size, dropout rate, optimizer, and activation function. The visual backbones were initialized with ImageNet weights, and the classification head was fine-tuned to adapt the features to our task.

B.2 Multimodal Hyperparameters

The selected hyperparameters are summarized in Table B.3.

Hyperparameter	Im	Ib	Ia	E3m	E7m
Dropout rate	0.5	0.5	0.5	0.5	0.5
Optimizer	Adam	Adam	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001	0.001	0.001
Epochs	16	16	20	16	16
Batch size	32	32	32	32	32

Table B.3: Hyperparameters used for training the multimodal models, where Im, Ib, Ia, E3m, and E7m denote InceptionNet + mBERT, InceptionNet + BERT, InceptionNet + AraBERT, EfficientNetB3 + mBERT, and EfficientNetB7 + mBERT, respectively

C Error Analysis

Both quantitative and qualitative error analyses were carried out to better understand the strengths and weaknesses of the best-performing model.

C.1 Quantitative Analysis

Table C.1 presents the analysis of our model and shows that it achieved an overall accuracy of 85%, with particularly strong results for the non-hateful class (F1-score of 0.91). For the hateful class, which represented the minority category, precision was lower at 0.36, though recall remained relatively high at 0.65. Given the dataset's class imbalance, macro F1-score (0.69) was used as the primary evaluation metric, as it equally weights both classes and provides a balanced performance measure. Table C.2 shows the official leaderboard, the system maintained strong generalization, with a macro F1-score of 63% on the unseen test set.

Class	Precision	Recall	F1-score	Support	G_score
non_hateful	0.96	0.87	0.91	281	0.91
hateful	0.36	0.65	0.46	31	0.48
macro avg	0.66	0.76	**0.69**	312	0.69
weighted avg	0.90	0.85	0.87	312	0.87
accuracy	-	-	0.85	312	-

Table C.1: Class-wise performance report on the validation set of the best-performing model

The confusion matrix in Figure C.1, provides a clear view of how the model distinguishes between hateful and non-hateful memes.The model correctly identified 267 non-hateful memes and 4 hateful memes. However, it misclassified 14 non-hateful memes as hateful and failed to detect 27 hateful memes.From a qualitative perspective, the fusion approach performs well when both text and image contribute useful and complementary information. In such cases, subtle textual hints combined with strong visual signals enable the model to correctly identify hateful content, where unimodal baselines often fail.

Position	Team_Name	Macro F1-Score (%)
1	NYUAD	80
2	yassirea	75
3	mzaytoon	74
4	itbaan	72
5	annasshaikh2003	68
6	**joy_2004114**	**63**

Table C.2: Leaderboard standings for the task

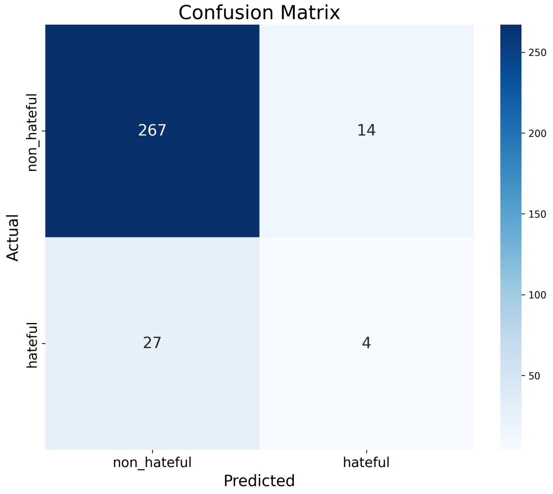


Figure C.1: Confusion matrics of best model

However, the model is not flawless. When one modality introduces misleading or ambiguous information, the fusion method can still occasionally succeed, but it is also vulnerable to misclassification. These errors highlight that while multimodal fusion strengthens overall performance, it remains sensitive to noise or imbalance in either modality.

C.2 Impact of Class Imbalance

The dataset's strong class imbalance (only ∼10% hateful samples) impacted the model's ability to maintain high precision for the hateful class. Although our model improved hateful recall compared to several baselines, targeted class balancing or augmentation strategies could further enhance performance in future work.

C.3 Qualatitative Analysis

Figure presents sample outputs from the model, illustrating both correct and incorrect classifications. In the first image, labeled as hateful, the model predicted non-hateful, likely due to its focus on explicit textual features while overlooking subtle visual cues indicating offensive intent. In the second image, labeled as non-hateful, the model misclassified it as hateful, reflecting sensitivity to certain visual or textual patterns that resemble hateful content. A major factor contributing to these errors is the significant class imbalance in the dataset, which biases the model toward dominant classes and limits its ability to generalize to underrepresented hateful samples. These findings highlight the need for improved context-aware multimodal modeling and strategies to mitigate imbalance effects.

Quasar at MAHED Shared Task : Decoding Emotions and Offense in Arabic Text using LLM and Transformer-Based Approaches

Md Sagor Chowdhury, Adiba Fairooz Chowdhury
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2004010, u2004014}@student.cuet.ac.bd

Abstract

The escalating presence of propaganda and hate speech on social media platforms underscores the need for robust automated detection systems to preserve the integrity of public discourse. Our team, participated in the MAHED 2025 Shared Task at the ArabicNLP 2025 conference, co-located with EMNLP 2025, focusing on Subtask 1 (Text-based Hate and Hope Speech Classification) and Subtask 2 (Emotion, Offensive, and Directed Hate Detection) in Arabic content. In Subtask 1, we experimented with models including XLM-RoBERTa-Large, Davlan/xlm-roberta-base-finetuned-arabic, asafaya/bert-base-arabic, aubmindlab/bert-base-arabertv2, Google Gemma-7B, and Qwen2.5-14B-Instruct, achieving the highest macro-f1 of 0.674 with Gemma-7B and ranking 12th on the leaderboard. In Subtask 2, using models such as aubmindlab/bert-base-arabertv2, Google Gemma-7B, Qwen2.5-14B-Instruct, asafaya/bert-base-arabic, and domain-specific hate-speech models, our best macro-f1 was 0.48 with both Gemma-7B and aubmindlab/bert-base-arabertv2, placing us 6th in the leaderboard.

1 Introduction

Hate and hope speech uses negative or positive expressions in text to influence readers' behavior, opinions, or emotions for a specific agenda. It is widespread on social media in tweets, posts, and comments, often with inherent bias. These speeches shape public perception and attract attention by amplifying offensive content, emotional appeals, or harmful narratives. Detecting hate, hope, and offensive content is essential to curb misleading or harmful information. Hate, hope, and offensive speech detection in Arabic text is challenging due to subtle sentiment, sarcasm, and context-dependent meanings. Social media content includes slang, abbreviations, and mixed styles, adding complexity. There is a gap in large-scale annotated datasets and specialized NLP tools for this compared to general sentiment analysis. This paper aims to detect such speech in Arabic social media posts and comments.

The MAHED 2025 shared task (Zaghouani et al., 2025) provides datasets for Subtask 1 and Subtask 2, labeled for offensive, hate, and hope speech, as a benchmark.We participated in subtask 1 and subtask 2. To achieve our goal, we augmented under-represented classes using back translation and evaluated models like XLM-RoBERTa-Large (Conneau et al., 2020), Davlan/xlm-roberta-base-finetuned-arabic (Davlan Team, 2023), asafaya/bert-base-arabic (Safaya et al., 2020), aubmindlab/bert-base-arabertv2 (Antoun et al., 2020), Google Gemma-7B (Mesnard et al., 2024) with classification head, and Qwen2.5-14B-Instruct (Yang et al., 2024). Each model was trained and assessed on the dataset. For Subtask 1 (Text-based Hate and Hope Speech Classification), Google Gemma-7B achieved a macro-F1 score of 0.674, ranking 12th. For Subtask 2 (Emotion, Offensive, and Directed Hate Detection), Gemma-7B and aubmindlab/bert-base-arabertv2 scored 0.48 macro-F1, placing 6th.

The core contributions of our research work include:

1. augmenting underrepresented classes using back translation,
2. leveraging external datasets to enrich training data, and
3. applying both large language models (LLMs) and Arabic-specific transformer models to improve detection of hate, hope, offensive, emotion, and directed hate speech in Arabic content.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 692–699
November 8-9, 2025 ©2025 Association for Computational Linguistics

Detailed implementation information is available in the linked GitHub repository [1]

2 Related Work

Hate speech detection in Arabic text has gained attention due to content moderation needs. Zaghouani et al. (2024a) developed a multi-label hate speech annotated Arabic dataset for model training. Biswas and Zaghouani (2025a) created an annotated corpus of Arabic tweets for hate speech analysis. This establishes benchmarks for Twitter-based systems. Hope speech serves as a counter to hate speech in research. Biswas and Zaghouani (2025b) introduced the EmoHopeSpeech dataset annotated for emotions and hope speech in English and Arabic. The bilingual dataset enables cross-lingual studies. It supports identifying positive and harmful content, offering a nuanced approach beyond binary classification. Hate speech research now includes multimodal analysis. Alam et al. (2024a) analyzed Arabic memes for propaganda-hate links using multi-agent LLMs. The ArMeme dataset (Alam et al., 2024b) shows how propagandistic memes evolve into hateful content. This highlights progression from persuasion to explicit hate. Propaganda detection intersects with hate speech. The WANLP 2022 shared task (Alam et al., 2022) set benchmarks for Arabic propaganda. Hasanain et al. (2024b) examined GPT-4 for propaganda spans in news. SemEval-2024 Task 4 (Dimitrov et al., 2024) focused on multilingual persuasion in memes. ArAIEval (Hasanain et al., 2023) targeted persuasion and disinformation in Arabic. Transformer models improve Arabic classification. Models like XLM-RoBERTa and AraBERT are standard. LLMs such as Gemma and Qwen perform well in tasks. Hasanain et al. (2024a) showed LLM effectiveness in propaganda annotation. Multitask learning detects emotions, offensive language, and hate using shared representations. Arabic hate speech detection continues to face challenges including dialectal variations, cultural context sensitivity, and evolving online hate speech patterns. The MAHED 2025 shared task builds upon these foundations while addressing contemporary challenges in Arabic social media content moderation through combining traditional classification with modern large language models.

3 Data

For Subtask 1, we used the dataset from the MAHED 2025 shared task (Zaghouani et al., 2025), consisting of Arabic social media posts labeled as *not_applicable*, *hope*, or *hate*, introduced in (Zaghouani et al., 2024b). The dataset is divided into training, development, and test sets, though specific split sizes are not detailed here. The training set includes 6,890 samples with notable imbalances: 3,697 *not_applicable*, 1,892 *hope*, and 1,301 *hate*, as shown in Figure 1.

For Subtask 2, we used the EmoHopeSpeech dataset (Zaghouani and Biswas, 2025), which contains 5,960 rows annotated with emotions, offensive and hate speech in English and Arabic.The distribution of frequecies for each label are given in Table A.

Examples of each label are given in section B

4 System

We have participated in subtask 1 and 2, which are an unimodal and multilabel text classification task. Figure 2 presents our proposed multi-output classification architecture for Arabic text analysis.

4.1 Data Augmentation

We have done data augmentation only for subtask-1.The original training dataset exhibited significant class imbalance with 3,697 not_applicable, 1,892 hope, and 1,301 hate instances. We implemented a multi-stage augmentation strategy to address this imbalance.

External Dataset Integration We incorporated additional datasets: 130 instances from an Arabic optimism dataset[2] for hope speech and additional hate speech samples from an external corpus[3].

Synonym Replacement and Back-Translation For the "hope" class, we applied Arabic synonym replacement using a comprehensive dictionary[4], preserving URLs, emojis, and punctuation while replacing content words. We generated 1,675 additional samples through this process. We further implemented back-translation (Arabic English Arabic) using Google Translate API: (1)

[1] GitHub Repository

[2] https://data.mendeley.com/datasets/mcnzzpgrdj/2

[3] https://www.sciencedirect.com/science/article/pii/S2352340923008144

[4] https://github.com/mdanok/ArabicLT/master/csv/synonyms.csv

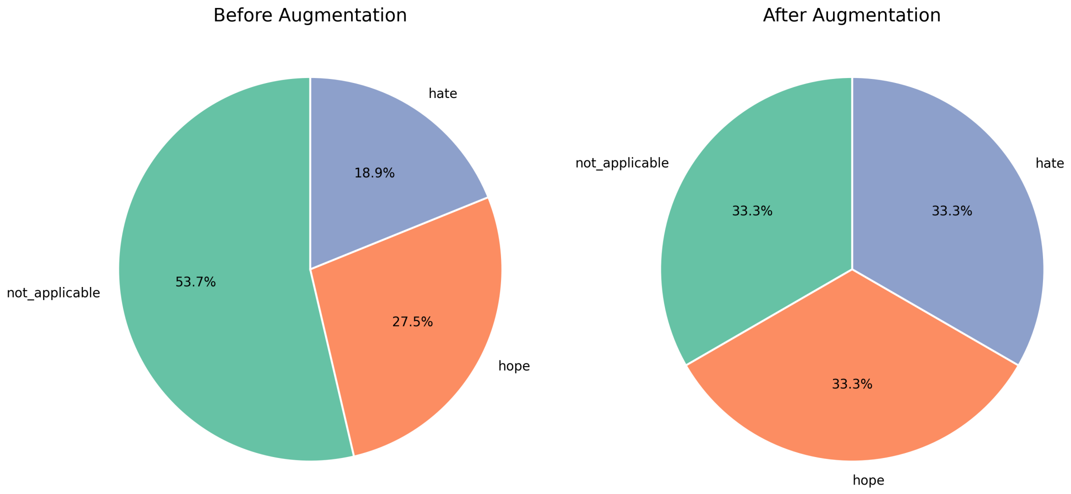


Figure 1: Label Distribution Before and After Augmentation for subtask-1

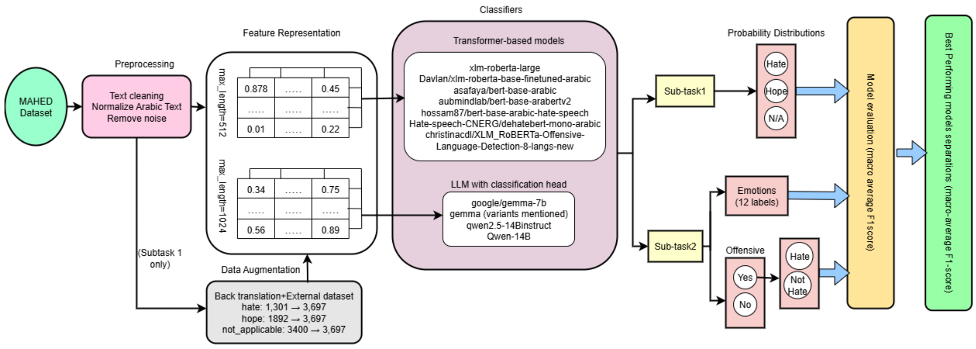

Classifiers
Feature Representation
Preprocessing
MAHED Dataset
Text cleaning
Normalize Arabic Text
Remove noise
max_length=512
0.878
.....
0.45
0.01
0.22
max_length=1024
0.34
0.75
0.56
0.89
Transformer-based models
xlm-roberta-large
Davlan/xlm-roberta-base-finetuned-arabic
asafaya/bert-base-arabic
aubmindlab/bert-base-arabertv2
hossam87/bert-base-arabic-hate-speech
Hate-speech-CNERG/dehatebert-mono-arabic
christinacdl/XLM_RoBERTa-Offensive-Language-Detection-8-langs-new
LLM with classification head
google/gemma-7b
gemma (variants mentioned)
qwen2.5-14Binstruct
Qwen-14B
(Subtask 1 only)
Data Augmentation
Back translation+External dataset
hate: 1,301 → 3,697
hope: 1892 → 3,697
not_applicable: 3400 → 3,697
Sub-task1
Sub-task2
Probability Distributions
Hate
Hope
N/A
Emotions (12 labels)
Offensive
Yes
No
Hate
Not Hate
Model evaluation (macro average F1score)
Best Performing models separations (macro-average F1-score)

Figure 2: Overview of our proposed multi-output classification system for Arabic text analysis.

translating synonym-replaced Arabic to English, (2) applying English synonym replacement using WordNet and spaCy on nouns, verbs, adjectives, and adverbs, and (3) translating back to Arabic. This introduced natural linguistic variations while preserving semantic content.

Label	**Before**	**After**
not_applicable	3,697	3,697
hope	1,892	3,697
hate	1,301	3,697
Total	**6,890**	**11,091**

Table 1: Dataset distribution before and after augmentation

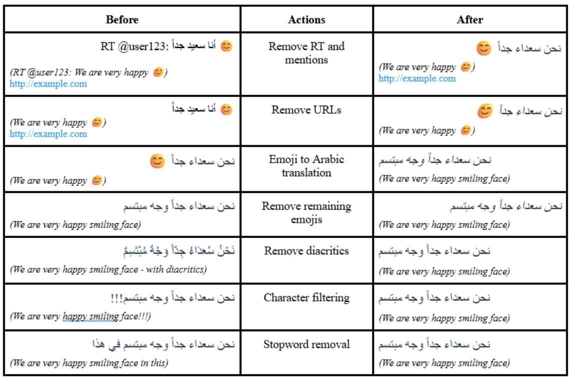
The augmentation successfully created a balanced dataset with 3,697 instances per class, representing a 61% increase in total samples and ensuring equal learning opportunities for all categories. Figure 1 illustrates the class distribution before and after the augmentation process.

4.2 Data Preprocessing

To ensure clean and consistent input for our models, we implemented a comprehensive preprocessing pipeline shown in this table2 for the Arabic text data. The preprocessing steps were applied sequentially to handle the specific challenges of Arabic social media text. This preprocessing pipeline ensured that our models received clean, normalized Arabic text optimized for classification tasks while preserving essential semantic content.

4.3 Initial Experimentation

In our initial experiments for both subtasks, we explored several transformer-based models to establish baselines and understand the performance landscape. Using `xlm-roberta-large` on both augmented+preprocessed and raw datasets, we observed that while the augmented data showed a strong bias towards the *hope* label, performance on the raw dataset was primarily skewed towards *not_applicable*. Balanced experiments on subsets demonstrated that data preprocessing



Before	Actions	After
RT @user123: أنا سعيد جداً 😊 *(RT @user123: We are very happy 😊)* http://example.com	Remove RT and mentions	نحن سعداء جداً 😊 *(We are very happy 😊)* http://example.com
أنا سعيد جداً 😊 *(We are very happy 😊)* http://example.com	Remove URLs	نحن سعداء جداً 😊 *(We are very happy 😊)*
نحن سعداء جداً 😊 *(We are very happy 😊)*	Emoji to Arabic translation	نحن سعداء جداً وجه مبتسم *(We are very happy smiling face)*
نحن سعداء جداً وجه مبتسم *(We are very happy smiling face)*	Remove remaining emojis	نحن سعداء جداً وجه مبتسم *(We are very happy smiling face)*
نَحْنُ سُعَدَاءُ جِدّاً وَجْهٌ مُبْتَسِمٌ *(We are very happy smiling face - with diacritics)*	Remove diacritics	نحن سعداء جداً وجه مبتسم *(We are very happy smiling face)*
نحن سعداء جداً وجه مبتسم!!! *(We are very happy smiling face!!!)*	Character filtering	نحن سعداء جداً وجه مبتسم *(We are very happy smiling face)*
نحن سعداء جداً وجه مبتسم في هذا *(We are very happy smiling face in this)*	Stopword removal	نحن سعداء جداً وجه مبتسم *(We are very happy smiling face)*

Table 2: Examples of Preprocessing Actions on Arabic Text.

and balancing played a crucial role in improving model performance. Building on this, we evaluated additional models including *Gemma*, *Qwen* (14B and 2.5-14B-instruct), *asafaya/bert-base-arabic*, *aubmindlab/bert-base-arabertv2*, and *Davlan/xlm-roberta-base-finetuned-arabic*, as well as specialized hate speech models such as *hossam87/bert-base-arabic-hate-speech* (Hossam, 2023) and *Hate-speech-CNERG/dehatebert-mono-arabic* (Aluru et al., 2020), which showed strong bias towards the *hate* label in training. Across these experiments, performances varied depending on model architecture and data preprocessing strategies.

4.4 Overview of the Adopted Model

For Subtask 1, we evaluated several models on different versions of the dataset, including XLM-RoBERTa-large, Qwen-14B, and Davlans XLM-RoBERTa-base fine-tuned models. Among these, Gemma7b with selected parameters(C) consistently achieved the highest accuracy across combinations of training, validation, and test sets, outperforming others with accuracies ranging from 0.47 to 0.67 depending on the dataset composition.

Similarly, for Subtask 2, Gemma was again chosen as the primary model. Other transformer-based models showed competitive performance, but Gemma provided the most consistent and reliable results for our multi-label Arabic classification task. We used the standard pre-trained tokenizer, set appropriate maximum sequence lengths, and experimented with hyperparameters such as learning rate, batch size, and number of epochs to optimize performance.

5 Results and Analysis

In this section, we summarize the key findings of our experiments, focusing on which approaches performed best rather than presenting full tables.

5.1 Evaluation Metrics

We evaluate using the macro-F1 score, which balances performance across all classes by averaging F1-scores independently for each label.

5.2 Comparative Analysis

Our experiments show that among all tested approaches, **Google Gemma-7B with DoRA configuration** achieved the best results for Subtask 1 (Hate and Hope Speech Classification), reaching an accuracy of **0.67**. The comprehensive performance comparison across different models and dataset configurations for Subtask 1 is presented in Table 3.

Model	Dataset Configuration	Macro-F1
XLM-RoBERTa-Large	Augmented + Preprocessed	0.33
	Given Dataset	0.54
	Given Dataset (1301 per label)	0.32
	Cleaned + 1301 per label + Non-cleaned Val	0.59
	Preprocessed + 1301 per label + Cleaned test	0.57
	Preprocessed given + Unprocessed test	0.23
Google Gemma-7B	Given + LoRA config	0.66
	Given + preprocessed test	0.60
	Given + 1301 per label	0.48
	Given + 1301 per label + processed test	0.47
	Given + DoRA + Unprocessed test	**0.67**
	Given + DoRA + processed test	0.64
Qwen-14B	1300 data samples	0.43
Davlan/xlm-roberta-base-arabic	Given Dataset	0.63
	Preprocessed Dataset	0.61
	Augmented + preprocessed	0.59

Table 3: Performance comparison for Subtask 1 across different models and dataset configurations.

For Subtask 2 (Emotion, Offensive, and Directed Hate Detection), the highest macro-F1 score (**0.48**) was obtained by three models: **Qwen2.5-14B-Instruct**, **aubmindlab/bert-base-arabertv2**, and **Google Gemma-7B**. The performance comparison for Subtask 2 is shown in Table 4.

6 Error Analysis

6.1 Confusion Matrix Analysis

To evaluate the classification performance in detail, we analyze the confusion matrices generated by our best performing Gemma-7B model. For Subtask 1, the confusion matrix (shown in Figure 8 in Appendix D) demonstrates the model's ability to distinguish between hate speech, hope speech, and not applicable content.

Model	Macro-F1	Notes
Qwen2.5-14B-Instruct	**0.48**	-
asafaya/bert-base-arabic (3 epochs)	0.45	-
asafaya/bert-base-arabic (20 epochs)	0.44	-
aubmindlab/bert-base-arabertv2	**0.48**	-
aubmindlab/bert-base-arabertv2	0.42	Preprocessed
Google Gemma-7B	**0.48**	-
Ensemble (XLM-RoBERTa + Gemma + dehatebert)	0.43	-

Table 4: Performance comparison for Subtask 2 showing macro-F1 scores.

For Subtask 2, we examine three separate confusion matrices corresponding to the multi-label classification components: emotion classification (Figure 9), offensive content detection (Figure 10), and hate speech detection within offensive content (Figure 11). These matrices provide insights into the model's performance across different aspects of the multi-label task.

6.2 Error Patterns

The confusion matrices reveal several key patterns in model performance:

- **Subtask 1 Performance**: The model shows good discrimination between hate and hope classes but occasionally confuses both with *not_applicable* content. This suggests that the model sometimes struggles to identify the presence of clear hate or hope indicators in ambiguous text.

- **Emotion Classification**: The model performs well on distinct emotions like joy and anger but struggles with subtle emotional distinctions. This indicates that while the model can capture clear emotional signals, it faces challenges in differentiating between closely related emotional states.

- **Offensive Content Detection**: The analysis shows high precision but some recall issues, particularly with borderline cases. This suggests the model tends to be conservative in its offensive content predictions, potentially missing some subtle forms of offensive language.

- **Hate Speech Detection**: Within offensive content, the model demonstrates the inherent challenge of distinguishing targeted hate from general offensive language. This highlights the complexity of the hate speech detection task, where the boundary between offensive and hateful content is often nuanced.

These error patterns provide valuable insights into the limitations of current approaches and suggest directions for future improvements in hate and hope speech classification systems.

7 Conclusion

Our study demonstrates the effectiveness of transformer-based and LLM approaches for Arabic hate, hope, offensive, and emotion detection, with Gemma-7B achieving the strongest results. However, the models show limitations in handling text with divine or religiously inspired speech, often misclassifying such hopeful expressions as neutral or humorous, as observed in our error analysis. Moreover, due to limited computational resources, we could not experiment with larger models capable of capturing broader context. As future work, we plan to incorporate domain-specific religious and cultural corpora to better model divine hopeful speech and explore larger-scale or more efficient models to enhance contextual understanding and overall robustness.

References

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmad, Md. Arid Hasan, and Maram Hasanain. 2024b. ArMeme: Propagandistic content in Arabic memes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21071–21090, Miami, Florida, USA. Association for Computational Linguistics.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the The Seventh WANLP*, Abu Dhabi, United Arab Emirates (Hybrid). ACL.

Sai Saket Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: A pre-trained arabic language model. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools*, Marseille, France. European Language Resource Association.

Md. Rafiul Biswas and Wajdi Zaghouani. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *CoRR*, abs/2505.11969.

Md. Rafiul Biswas and Wajdi Zaghouani. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *CoRR*, abs/2505.11959.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Davlan Team. 2023. Xlm-roberta base fine-tuned for arabic. No specific academic paper is associated with Davlan/xlm-roberta-base-finetuned-arabic; refer to the Hugging Face model page for details.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Maram Hasanain, Fatema Ahmad, and Firoj Alam. 2024a. Large language models for propaganda span annotation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14522–14532, Miami, Florida, USA. Association for Computational Linguistics.

Maram Hasanain, Fatema Ahmed, and Firoj Alam. 2024b. Can gpt-4 identify propaganda? annotation and detection of propaganda spans in news articles. In *Proceedings of the 2024 Joint International Conference On Computational Linguistics, Language Resources And Evaluation*, LREC-COLING 2024, Torino, Italy.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion techniques and disinformation detection in arabic text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hossam. 2023. Bert base arabic hate speech detection model. Hugging Face model repository.

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

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-based multi-label classification for offensive language detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Barcelona (online). International Committee for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024a. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

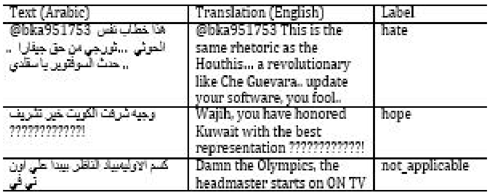
Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024b. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

A Label distribution

B Examples from dataset

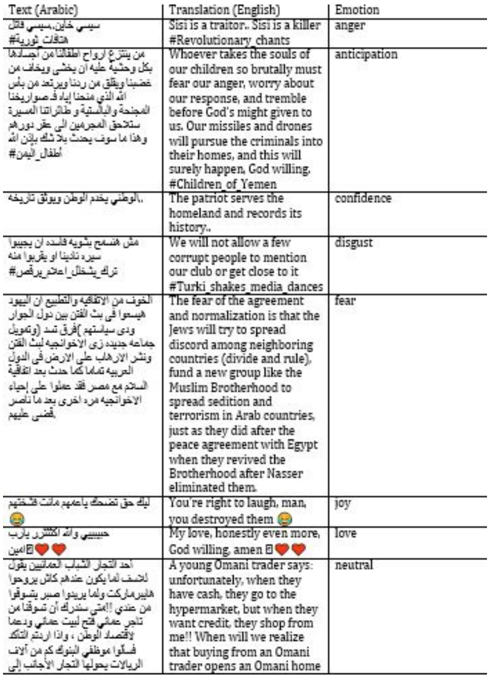
C Parameter Settings

For both Subtask 1 and Subtask 2, we adopted the DoRA-enhanced transformer model, "Gemma", and set the parameters as follows. The learning rate was set to 1×10^{-4} with no weight decay applied. The model was trained for 3



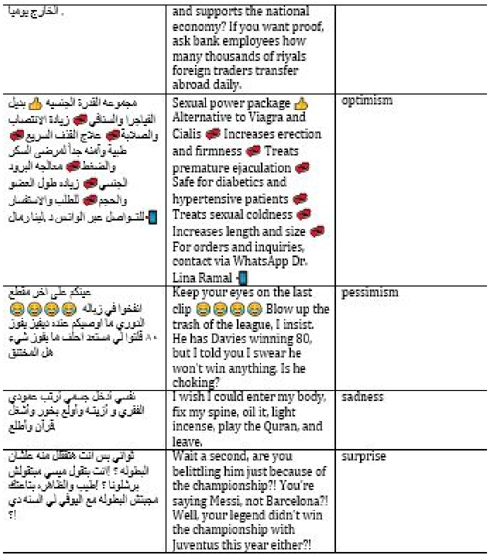
Text (Arabic)	Translation (English)	Label
@bka951753 هذا خطاب نفس الحوثي ...ثورجي من حق جيفارا .. حدث السوفتوير يا سقدي ..	@bka951753 This is the same rhetoric as the Houthis... a revolutionary like Che Guevara.. update your software, you fool..	hate
وجيه شرفت الكويت خير تشريف ????????????!	Wajih, you have honored Kuwait with the best representation ????????????!	hope
كسم الاولمبياد الناظر بيبدا على اون تي في	Damn the Olympics, the headmaster starts on ON TV	not_applicable

Figure 3: Data example for subtask 1



Text (Arabic)	Translation (English)	Emotion
سيسي خاين..سيسي قاتل #هتافات_ثورية	Sisi is a traitor.. Sisi is a killer #Revolutionary_chants	anger
من ينتزع ارواح اطفالنا من أجسادها بكل وحشية عليه ان يخشى ويخاف من غضبنا ويقلق من ردنا ويرتعد من بأس الله الذي منحنا إياه فـ صواريخنا المجنحة والبالستية و طائراتنا المسيرة ستلاحق المجرمين الى عقر دورهم وهذا ما سوف يحدث بلا شك بإذن الله #اطفال_اليمن	Whoever takes the souls of our children so brutally must fear our anger, worry about our response, and tremble before God's might given to us. Our missiles and drones will pursue the criminals into their homes, and this will surely happen, God willing. #Children_of_Yemen	anticipation
الوطني يخدم الوطن ويوثق تاريخه..	The patriot serves the homeland and records its history..	confidence
مش هنسمح بشويه فاسده ان يجيبوا سيرة نادينا او يقربوا منه #تركي_يشخلل_اعلام_يرقص	We will not allow a few corrupt people to mention our club or get close to it #Turki_shakes_media_dances	disgust
الخوف من الاتفاقيه والتطبيع ان اليهود هيسعوا فى بث الفتن بين دول الجوار ودى سياستهم)فرق تسد (وتمويل جماعة جديده زى الاخوانجيه لبث الفتن ونشر الارهاب على الارض فى الدول العربيه تماما كما حدث بعد اتفاقية السلام مع مصر فقد عملوا على إحياء الاخوانجيه مره اخرى بعد ما ناصر قضى عليهم.	The fear of the agreement and normalization is that the Jews will try to spread discord among neighboring countries (divide and rule), fund a new group like the Muslim Brotherhood to spread sedition and terrorism in Arab countries, just as they did after the peace agreement with Egypt when they revived the Brotherhood after Nasser eliminated them.	fear
ليك حق تضحك ياعمهم مانت فشختهم 😂	You're right to laugh, man, you destroyed them 😂	joy
حبيبي والله اكثرر يارب امين□❤❤	My love, honestly even more, God willing, amen □❤❤	love
أحد التجار الشباب العمانيين يقول للأسف لما يكون عندهم كاش يروحوا هايبرماركت ولما يريدوا صبر يتسوقوا من عندي !!متى سندرك أن تسوقنا من تاجر عماني فتح لبيت عماني ودعما لاقتصاد الوطن ، واذا اردتم التأكد فسألوا موظفي البنوك كم من آلاف الريالات يحولها التجار الأجانب إلى	A young Omani trader says: unfortunately, when they have cash, they go to the hypermarket, but when they want credit, they shop from me!! When will we realize that buying from an Omani trader opens an Omani home	neutral

Figure 4: Data example for emotion column of subtask 2

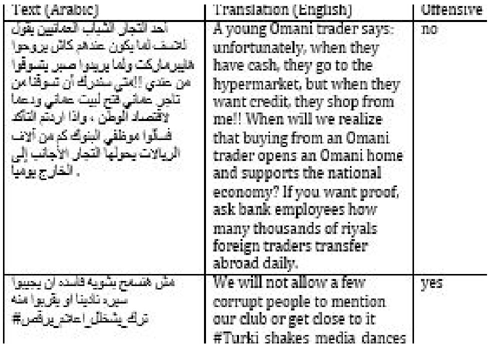


الخارج يوميا .	and supports the national economy? If you want proof, ask bank employees how many thousands of riyals foreign traders transfer abroad daily.	
مجموعه القدرة الجنسيه 👍 بديل الفياجرا والسنافي 💋 زيادة الانتصاب والصلابة 💋 علاج القذف السريع 💋 طبية وآمنه جداً لمرضى السكر والضغط 💋 معالجة البرود الجنسي 💋 زيادة طول العضو والحجم 💋 للطلب والاستفسار📲 التـواصل عبر الواتس د.لينا رمال	Sexual power package 👍 Alternative to Viagra and Cialis 💋 Increases erection and firmness 💋 Treats premature ejaculation 💋 Safe for diabetics and hypertensive patients 💋 Treats sexual coldness 💋 Increases length and size 💋 For orders and inquiries, contact via WhatsApp Dr. Lina Ramal 📲	optimism
عينكم على اخر مقطع 😂😂😂😂 انفخوا في زباله الدوري ما اوصيكم عنده ديفيز يفوز ٨٠ قلتوا لي مستعد احلف ما يفوز شيء هل المختنق	Keep your eyes on the last clip 😂😂😂😂 Blow up the trash of the league, I insist. He has Davies winning 80, but I told you I swear he won't win anything. Is he choking?	pessimism
نفسي ادخل جسمي أرتب عمودي الفقري و أزيته وأولع بخور وأشغل قرآن واطلع	I wish I could enter my body, fix my spine, oil it, light incense, play the Quran, and leave.	sadness
ثواني بس انت هتقلل منه علشان البطوله ؟ انت بتقول ميسي مبتقولش برشلونا ؟ إطيب والظاهر بتاعتك مجبتش البطوله مع اليوفي لي السنه دي ؟!	Wait a second, are you belittling him just because of the championship?! You're saying Messi, not Barcelona?! Well, your legend didn't win the championship with Juventus this year either?!	surprise

Figure 5: More example for emotion column of subtask 2

Table 5: Label distribution in the EmoHopeSpeech dataset (subtask2)

Column name	Label	Frequency
Emotion	Anger	1551
	Disgust	777
	Neutral	661
	Love	593
	Joy	533
	Anticipation	491
	Optimism	419
	Sadness	335
	Confidence	210
	Pessimism	194
	Surprise	143
	Fear	53
Offensive	No	4216
	Yes	1744
Hate (if Offensive = Yes)	Not_hate	1431
	Hate	303

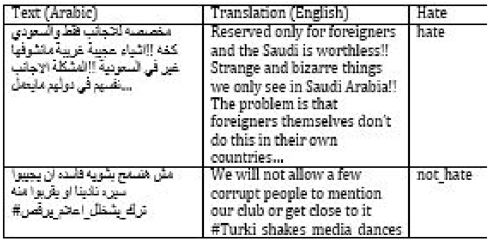


Text (Arabic)	Translation (English)	Offensive
أحد التجار الشباب العمانيين يقول للأسف لما يكون عندهم كاش يروحوا هايبرماركت ولما يريدوا صبر يتسوقوا من عندي !!متى سندرك أن تسوقنا من تاجر عماني فتح لبيت عماني ودعما لاقتصاد الوطن ، واذا اردتم التأكد فسألوا موظفي البنوك كم من آلاف الريالات يحولها التجار الأجانب إلى الخارج يوميا .	A young Omani trader says: unfortunately, when they have cash, they go to the hypermarket, but when they want credit, they shop from me!! When will we realize that buying from an Omani trader opens an Omani home and supports the national economy? If you want proof, ask bank employees how many thousands of riyals foreign traders transfer abroad daily.	no
مش هنسمح بشويه فاسده ان يجيبوا سيرة نادينا او يقربوا منه #تركي_يشخلل_اعلام_يرقص	We will not allow a few corrupt people to mention our club or get close to it #Turki_shakes_media_dances	yes

Figure 6: Data example of offensive column of subtask 2

epochs, with a per-device training batch size of 1 and gradient accumulation over 4 steps to simulate a larger batch size. Warmup steps were set to 10, and the optimizer used was `paged_adamw_8bit`. We enabled mixed precision training with bf16 for efficiency. The maximum sequence length for tokenization was 1024, and padding was applied dynamically using the `DataCollatorWithPadding`.

Model checkpoints were saved every 50 steps,



Text (Arabic)	Translation (English)	Hate
مخصصه للاجانب فقط والسعودي كخه !!اشياء عجيبة غريبة ماتشوفها غير في السعودية !!المشكلة الاجانب نفسهم في دولهم مايعمل...	Reserved only for foreigners and the Saudi is worthless!! Strange and bizarre things we only see in Saudi Arabia!! The problem is that foreigners themselves don't do this in their own countries...	hate
مش هنسمح بشويه فاسده ان يجيبوا سيرة نادينا او يقربوا منه #تركي_يشخلل_اعلام_يرقص	We will not allow a few corrupt people to mention our club or get close to it #Turki_shakes_media_dances	not_hate

Figure 7: Data example of hate column of subtask 2

and logging was performed every 10 steps. Unused columns in the dataset were removed to optimize memory usage. The DoRA configuration was applied with a rank r of 4, LoRA alpha of 32, LoRA dropout of 0.1, and targeting the projection layers `q_proj` and `v_proj`.

D Confusion Matrices

This appendix presents the confusion matrices for both subtasks, providing detailed visualization of the classification performance.

D.1 Subtask 1: Hate and Hope Speech Classification

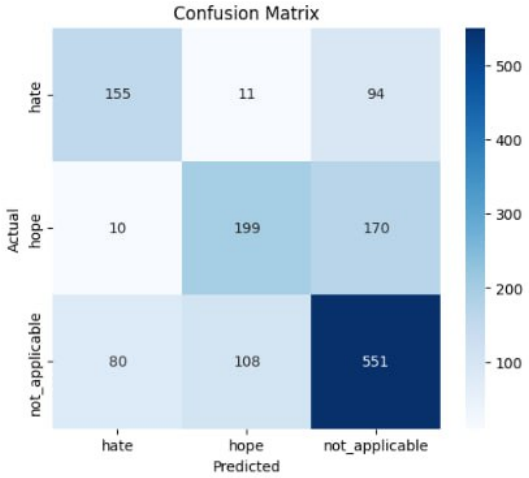


Figure 8: Confusion matrix for Subtask 1 (Hate and Hope Speech Classification) using Gemma-7B model.

D.2 Subtask 2: Multi-label Classification

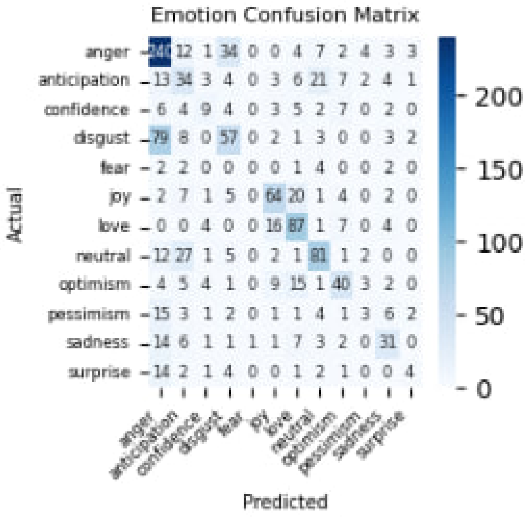


Figure 9: Confusion matrix for Emotion classification in Subtask 2.

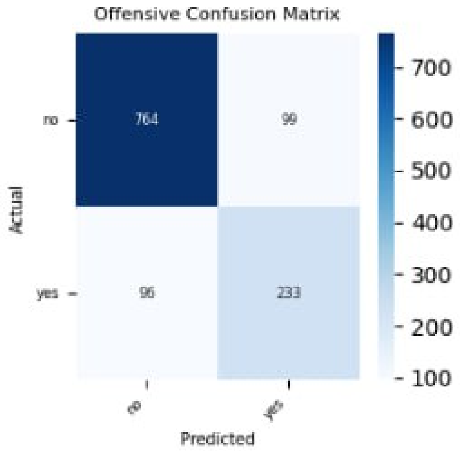


Figure 10: Confusion matrix for Offensive content detection in Subtask 2.

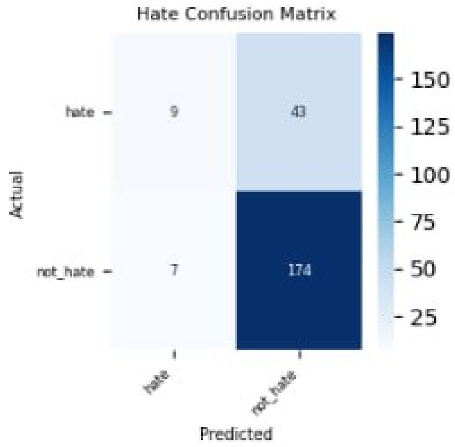


Figure 11: Confusion matrix for Hate speech detection in Subtask 2.

CUET_Zahra_Duo@Mahed 2025: Hate and Hope Speech Detection in Arabic Social Media Content using Transformer

Walisa Alam, Mehreen Rahman, Shawly Ahsan and Mohammed Moshiul Hoque
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2004015, u2004033, 22mcse105}@student.cuet.ac.bd, moshiul_240@cuet.ac.bd

Abstract

In recent years, online social life has become an integral part of the global landscape, with social media platforms enabling users to express a wide range of emotions and opinions. In the Arabic-speaking world, navigating the dual nature of content—encompassing both hate and hope speech—remains challenging due to linguistic and cultural complexities. The MAHED 2025 shared task at ArabicNLP 2025 addressed this by focusing on detecting both hate and hope speech in Arabic social media. This paper describes our approach for subtask 1, utilizing various machine learning, deep learning, and transformer models for classification. AraBERT-large-v2 yielded the highest macro f_1-score of 0.698, earning 8^{th} place on the leaderboard.

1 Introduction

Social media platforms, such as Facebook and Twitter, enable widespread communication but also accelerate the spread of hate speech, which can fuel hostility and deepen social divides. The hateful content often spreads farther and faster than non-hateful material, mainly due to closely connected online communities (Mathew et al., 2019). In Arabic-speaking contexts, detecting hate speech is challenging due to dialectal diversity, frequent code-switching, rich morphological structures, orthographic variation, and cultural nuances (Elmadany et al., 2024).

Transformer architectures have significantly advanced hate speech detection. Recent research has shifted from traditional and deep learning models to transformer-based approaches, including BERT and its multilingual variants. Although these models achieve state-of-the-art performance, they also introduce higher computational costs, algorithmic biases, data scarcity, and inconsistent evaluation practices. The **MAHED 2025** shared task (Zaghouani et al., 2025) focuses on detecting hope and hate emotions in Arabic content. This study addresses subtask 1, which involves classifying hate and hope speech in Arabic texts. The primary contributions of this work are as follows:

- Investigated the efficacy of various machine learning models (Logistic Regression, Decision Tree, Random Forest, Naive Bayes, MNB, KNN, and XGBoost), deep learning models (CNN, BiLSTM, and CNN-BiLSTM), and transformer-based models (MARBERT, AraBERT-base, and AraBERT-large) in detecting both hate and hope speech in Arabic texts.
- Presented a transformer-based approach using **AraBERT-large** to classify Arabic social media texts into *hate*, *hope*, and *not_applicable* categories.

2 Related Work

Extensive research has been conducted on hate and hope speech detection, ranging from classical ML to DL and from transformer models to large language models (LLMs). Roy et al. (2022) applied classical ML models, using Logistic Regression and TF-IDF features, Random Forest, and XGBoost. Their best-performing model was Random Forest, with an F1 score of 0.96. Yang et al. (2023) used several LLMs like GPT-3.5-turbo-0613, Flan-T5, T5-large, GPT-2-large, and two variants of the HARE framework, Fr-HARE and Co-HARE, to improve accuracy. Among the models tested, Co-HARE with Flan-T5 (large) achieved the highest accuracy. Usman et al. (2025) addresses multilingual hate speech detection in English, Urdu, and Spanish using a trilingual dataset of 10,193 tweets. The evaluated models include LLMs (GPT-3.5 Turbo, Qwen 2.5 72B), transformers (BERT, RoBERTa), and SVM with TF-IDF features. Qwen 2.5 72B achieved the best performance overall, especially in the joint multilingual setting.

Recent research has seen significant advancements in the detection of Arabic hate and hope speech. Zaghouani et al. (2024) evaluated LR, RF, Gradient Boosting, SVM, Decision Tree, and AraBERT for this task. AraBERT was the best-performing model, with an accuracy of 0.83. Charfi et al. (2024) introduced the ADHAR dataset covering various dialects. Using AraBERT, they achieved high performance in hate speech detection (F1 score of 0.95). Alghamdi et al. (2024) presented AraTar, where AraBERTv0.2 (base) achieved the best performance. Yagci et al. (2024) worked on Turkish and Arabic hate speech detection in the HSD-2Lang shared task, using AraBERTv02-Twitter fine-tuned with the AdamW optimizer. Their best-performing configuration achieved 0.89 accuracy and 0.74 F1 score for Arabic texts. AlDahoul and Zaki (2025) addressed Arabic hate and hope speech detection, where an ensemble of GPT-4o-mini, Gemini Flash 2.5, and Google text embedding with SVM, combined with a fine-tuned GPT-4o-mini hope/not classifier, achieved the best performance (macro-F1 score of 72.1%).

Previous research on Arabic hate and hope speech detection has been constrained by limited data. In this study, we overcame these constraints by implementing improved data cleaning and augmentation strategies.

3 Task and Dataset Distribution

The MAHED 2025 shared task aims to advance research on detecting hate speech, hope speech, and emotional expressions in Arabic content (Zaghouani et al., 2024; Zaghouani and Biswas, 2025b,a). We participated in subtask 1, which involved classifying Arabic texts into three categories:

- **Hate:** Text expressing hostility, bias, or discrimination against certain individuals or groups.
- **Hope:** Text that communicates positivity, encouragement, or supportive messages.
- **Not Applicable:** Text that does not contain elements of hate or hope speech.

The dataset consists of text samples collected from Arabic social media platforms and is divided into a training set (T_{trn}), validation set (T_{val}) and test set (T_{tst}). Table 1 shows the statistics of the dataset.

Table 1: Dataset statistics, where T_w, T_{uw}, T_{mw}, and T_{aw} indicate the number of total words, unique words, maximum words per text, and average words per text in the training set, respectively.

Attributes	Hate	Hope	N /A	Total
T_{trn}	1,301	1,892	3,697	6,890
T_{val}	261	409	806	1,476
T_{tst}	287	422	768	1,477
T_w	30,855	41,317	82,700	1,54,872
T_{uw}	15,606	22,499	42,212	68,126
T_{mw}	92	105	107	–
T_{aw}	23.0	21.0	22.0	–

4 Methodology

This study explores several ML, DL, and transformer-based architectures. As shown in Figure 1, the adopted model follows a multi-stage design.

4.1 Data Preprocessing

The text preprocessing pipeline systematically cleans Arabic tweets to improve the model performance. It removes punctuation (including Arabic symbols), numbers, Latin letters, emojis, and extra whitespace, converts the text to lowercase, and normalizes the Arabic script by removing Tashkeel. Additionally, tweet-specific elements, such as URLs, mentions, and hashtags, were handled, and informal text was converted to standard Arabic. This preprocessing ensures that the input data is normalized and noise-free, making it suitable for ML, DL, and transformer-based models.

4.2 Data Augmentation

We employed contextual word embedding–based augmentation using the Arabic BERT model (Antoun et al., 2020) via the *nlpaug*[1] library, where selected words were replaced with contextually similar alternatives predicted by the model. This approach preserves the semantic meaning of the original text while introducing lexical and structural variations, ensuring that the augmented samples retain their original classification labels.

4.3 Overview of the Adopted Model

We adopted ML, DL, and transformer-based classifiers for Arabic hate and hope speech detection.

4.3.1 ML Models

For feature representation, we employed the TF-IDF scheme to represent the textual data. Using the

[1] https://github.com/makcedward/nlpaug

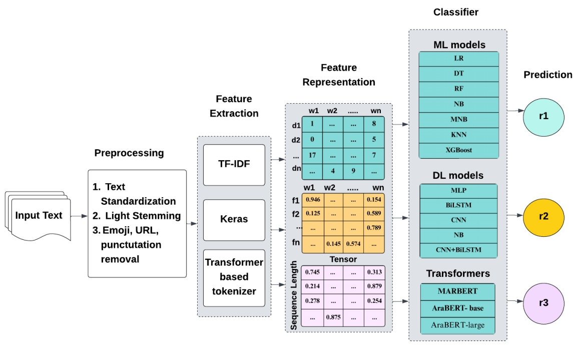


Figure 1: Overview of the adopted model

TF-IDF features, we evaluated several ML classifiers, including Logistic Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost (XGB). LR was trained with a maximum of 1200 iterations, while DT was trained with its default configuration. RF was optimized with tuned estimators and depth, and SVM was employed with a linear kernel. For XGBoost, we applied '*multi:softprob*' objective, 100 rounds for boosting, multiclass logloss for evaluation, and '*gpu_hist*' for the tree construction algorithm. The KNN model was trained with 12 neighbours.

4.3.2 DL Models

For the DL models, the text was first tokenized using the Keras library[2] with a maximum vocabulary size of 10,000 words, and sequences were padded or truncated to a fixed length of 150 tokens. Multiple neural network architectures were implemented, including Multi-Layer Perceptron (MLP) with TF-IDF inputs, BiLSTM, CNN, and CNN+BiLSTM. MLP was configured with 3 layers (512, 256, 256), ReLU activation, softmax output, and trained for 150 epochs with a batch size of 64 and early stopping. The BiLSTM model consisted of 512 units, a dropout rate of 0.2, and a dense layer of 256. It was trained for 150 epochs with a batch size of 64 and learning rate decay (0.96, 1000). The CNN architecture applied convolution layers with 512 and 256 filters, a kernel size of 5, vocabulary size of 10,000, maximum input length of 150, dropout rate of 0.2, and early stopping. The hybrid CNN+BiLSTM combined a 512-filter convolution layer with a 256-unit BiLSTM layer, followed by a dense layer of 128 units and a dropout rate of 0.2. The CNN+BiLSTM model was trained for 100 epochs with a batch size of 64 and early stopping. All models employed embedding layers and a softmax function for multiclass classification.

4.3.3 Transformer-Based Models

For each transformer-based model, the texts were tokenized and padded using their respective tokenizers from the HuggingFace library. We employed several transformer-based models for Arabic text classification, including AraBERT-base (Safaya et al., 2020), MARBERT (Abdul-Mageed et al., 2021), and AraBERT-large (Antoun et al., 2020). Each transformer comprises multiple encoder layers with multi-head self-attention, feedforward networks, residual connections, and layer normalization, with dropout applied to the hidden states and attention weights to prevent overfitting. The contextual representation of the [CLS] token was fed into a fully connected layer for classification into three categories (hope, hate, and not applicable). AraBERT-base and MARBERT were trained with a learning rate of 1×10^{-5} for 20 epochs with a batch size of 128, while AraBERT-large was trained with varying learning rates (3.5×10^{-6} to 2×10^{-5}), epochs, and batch sizes (128 and 256), with or without augmentation.

5 Result Analysis

All experiments were conducted on Kaggle using two NVIDIA Tesla T4 GPUs with 16 GB of GPU memory each and 32 GB system RAM. We evaluated the performance of the models using several metrics, including precision, recall, and macro f_1 score (MF1). MF1 was chosen as the primary metric to ensure a balanced performance evaluation of the models. Table 2 presents a comparative analysis of the performance achieved by ML, DL, and transformer-based models for Arabic text-based classification of hope and hate speech.

Among the ML classifiers, Naive Bayes performed best, likely because its probabilistic nature enhanced its ability to predict positive outcomes, boosting Recall and thus MF1, which is especially suitable in cases where positive instances are harder to capture. In the DL category, CNN + BiLSTM performed best, with an MF1 score of 0.619, because it effectively integrated CNN's local feature extraction with BiLSTM's sequential context modeling, resulting in a stronger precision–recall balance. However, AraBERT-large was the standout performer among the AraBERT family. Outper-

[2] https://keras.io/

Table 2: Performance comparison of ML, DL, and transformer-based models for Arabic hate and hope speech classification.

Model	Precision	Recall	MF1
LR	0.652	0.519	0.542
DT	0.516	0.498	0.506
RF	0.652	0.492	0.509
NB	0.638	0.541	0.563
MNB	0.789	0.381	0.325
KNN	0.597	0.494	0.513
XGBoost	0.633	0.517	0.538
MLP	0.617	0.554	0.581
BiLSTM	0.634	0.565	0.582
CNN	0.598	0.594	0.607
CNN + BiLSTM	0.601	0.623	0.619
MARBERT	0.640	0.640	0.640
AraBERT-base	0.600	0.600	0.600
AraBERT-large	**0.694**	**0.697**	**0.695**

forming MARBERT and AraBERT-base in Precision, Recall, and MF1 scores, AraBERT-large emerged as the top variant with the highest scores across all metrics, achieving an MF1 score of 0.695. This superior performance can be attributed to AraBERT-large's broader contextual coverage and stronger capacity to capture the morphological richness of Arabic, which enabled it to outperform the smaller models.

5.1 Ablation Study

The results of the ablation study on the classification of hatred and hope discourse in Arabic are shown in Table 3, with distinct reports for the development and testing stages of the models. The best-performing model was trained for a maximum of 20 epochs, incorporating early stopping with a patience of 4. The model converged after 12 epochs. In the development phase, the batch size, learning rate, and sequence length had a clear effect. A batch size of 8 and a learning rate of 1×10^{-5} led to stable training, whereas increasing the rate to 2×10^{-5} slightly reduced performance. Preprocessing combined with augmentation helped AraBERT-large and MARBERT achieve the highest MF1 score of 0.64, whereas raw data or preprocessing alone yielded lower scores.

In the testing phase, AraBERT-large with preprocessing achieved the best MF1 score (0.69). Using a smaller learning rate of 3.5×10^{-6} and a longer sequence length of 256 improved generalization, highlighting the importance of careful hyperparameter tuning along with preprocessing.

5.2 Error Analysis

A detailed error analysis was carried out to understand the performance of the best-performing model (i.e., AraBERT-large).

5.2.1 Quantitative Error Analysis

The results highlight the strong performance of the AraBERT-large model in classifying Arabic social media texts into hate and hope categories. The confusion matrix shown in Figure 2 provides a quantitative breakdown of the predictions.

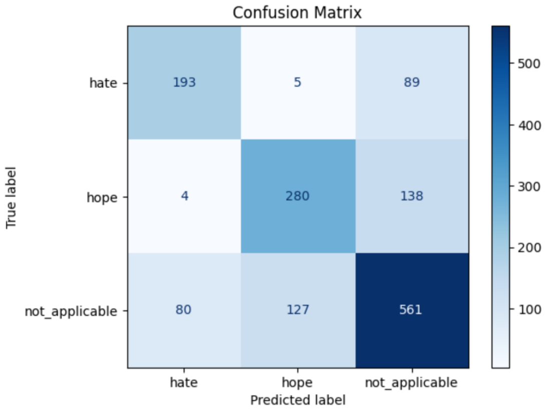


Figure 2: Confusion matrix of the AraBERT-large model for the test set

The analysis showed that the model successfully identified 193 hate samples, 280 hope samples, and 561 not_applicable samples. However, there were a few misclassifications, with 5 hate instances incorrectly labeled as hope and 4 hope instances misclassified as hate. A larger source of error comes from confusion with the not_applicable class, where 89 hate and 138 hope samples are wrongly predicted as not_applicable, whereas 80 not_applicable samples are labeled as hate and 127 as hope. These errors can be attributed to overlapping linguistic cues across categories and the class imbalance.

5.2.2 Qualitative Error Analysis

Table 4 demonstrates some sample predictions made by the AraBERT-large model. Here, samples 2 and 3 were correctly classified, whereas samples 1, 4, and 5 were misclassified. Sample 1 was mislabelled as not_applicable when it was hate due to sarcasm diluting the hateful signals. Sample 4 was predicted as hate instead of not_applicable because of the strong offensive words that the model associates with hate, and Sample 5 was inaccurately classified as hope instead of not_applicable because the positive and uplifting tone resembles

Table 3: Ablation study on the impact of hyperparameters on the performance of the transformer-based models.

Model	Method	Batch	LR	Epochs	MaxLen	MF1
Development Phase						
AraBERT-base	Preproc + Aug	8	1×10^{-5}	20	128	0.60
MARBERT	Preproc + Aug	8	1×10^{-5}	20	128	0.64
AraBERT-large	Raw data	8	1×10^{-5}	20	128	0.61
AraBERT-large	Preprocessed	8	1×10^{-5}	20	128	0.62
AraBERT-large	Preproc + Aug	8	1×10^{-5}	20	128	0.64
AraBERT-large	Preproc + Aug	8	2×10^{-5}	3	128	0.63
Testing Phase						
AraBERT-large	Raw data	8	1×10^{-5}	20	128	0.67
AraBERT-large	Preproc + Aug	8	3.5×10^{-6}	20	256	0.69
AraBERT-large	**Preprocessed**	**8**	3.5×10^{-6}	**20**	**256**	**0.69**

the hope class. These nuances highlight the importance of qualitative analysis in understanding the model performance in specific cases. Moreover, we observed that dialectal words introduce challenges in the detection of hate and hope speech. Sample 1 includes the Gulf/Yemeni dialect expression "ba'aysh" ("with what") and sample 5 contains the Egyptian/Levantine colloquial word "teslamy" ("thank you" / "bless you"). The presence of dialectal expressions in these samples underscores the complexity of accurately classifying texts in diverse Arabic dialects.

Table 4: Sample output predictions by the AraBERT-large model, where Arabic texts were translated using Google Translate.

Sample Text	Actual	Predicted
@shatat_20 @zainabera نعوضكم يالعراكي بس انقلع قم يعني بأيش (@shatat_20 @zainabera How can we compensate you? Just get out of here, you brat)	hate	N/A
@Mp_M_Alhazmi هو كا الشيعه دين والخرافه الشرك (@Mp_M_Alhazmi The Shiite religion is polytheism and superstition.)	hate	hate
الشعب... ايها معكم ..مواصلون الخير مساء عمانتل من الجماعيه الهجره 51 (Good evening.. We continue with you, dear people... The mass migration from Omantel 51)	hope	hope
وجوهكم تتبدل خنازير منكم اوسخ يوجد لا والنظام داعش مقولاتكم اين البرق بسرعه اجل فيها تستنجد ليش واحده لعمله وجهان الان؟ (There is no one dirtier than you pigs, your faces change at lightning speed. Where are your sayings? ISIS and the regime are two sides of the same coin. Why are you seeking their help now?)	N/A	hate
.... الاوليمبياد في مصر فخر سمير ساره تسلمي (Sarah Samir, Egypt's pride in the Olympics... Thank you)	N/A	hope

6 Conclusion

This work evaluated multiple ML, DL, and transformer-based models for detecting hate and hope speech in Arabic. The AraBERT-large model demonstrated the highest performance, achieving a macro f_1 score of 0.69 and surpassing all other models tested, benefiting from its broader contextual coverage and stronger ability to capture the morphological richness of the Arabic language. However, the system is limited by class imbalance, challenges in capturing nuanced or context-dependent meanings, and its dependence on the quality of the training data. Future work should focus on augmenting the dataset to mitigate class imbalance, integrating multilingual or cross-domain data, and investigating hybrid-model architectures to enhance predictive accuracy.

Limitations

The developed model has several limitations. It relies solely on textual input and cannot leverage multimodal signals, such as images or videos that often accompany social media posts. Its performance is also sensitive to preprocessing and augmentation strategies, which may not generalize well to unseen data. Moreover, training on a single dataset introduces the risk of bias and limits the model's adaptability to other dialects, domains, and code-switched texts.

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.

Nouar AlDahoul and Yasir Zaki. 2025. Detecting hope, hate, and emotion in arabic textual speech and multi-modal memes using large language models. *Preprint*, arXiv:2508.15810.

Seham Alghamdi, Youcef Benkhedda, Basma Alharbi, and Riza Batista-Navarro. 2024. AraTar: A corpus to support the fine-grained detection of hate speech targets in the Arabic language. 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 1–12, Torino, Italia. ELRA and ICCL.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Anis Charfi, Mabrouka Bessghaier, Raghda Akasheh, Andria Atalla, and Wajdi Zaghouani. 2024. Hate speech detection with adhar: a multi-dialectal hate speech corpus in arabic. *Frontiers in Artificial Intelligence*, 7.

Ahmed Elmadany, Wajdi Zaghouani, and Nizar Habash. 2024. A survey on arabic natural language processing: Challenges and applications. *ACM Computing Surveys*, 57(2):1–40.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science (WebSci '19)*, pages 173–182, Boston, MA, USA. ACM.

Pradeep Roy, Snehaan Bhawal, Abhinav Kumar, and Bharathi Raja Chakravarthi. 2022. IIITSurat@LT-EDI-ACL2022: Hope speech detection using machine learning. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 120–126, Dublin, Ireland. Association for Computational Linguistics.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Muhammad Usman, Muhammad Ahmad, M. Shahiki Tash, Irina Gelbukh, Rolando Quintero Tellez, and Grigori Sidorov. 2025. Multilingual hate speech detection in social media using translation-based approaches with large language models. *Preprint*, arXiv:2506.08147.

Utku Yagci, Egemen Iscan, and Ahmet Kolcak. 2024. ReBERT at HSD-2Lang 2024: Fine-tuning BERT with AdamW for hate speech detection in Arabic and Turkish. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 195–198, St. Julians, Malta. Association for Computational Linguistics.

Yongjin Yang, Joonkee Kim, Yujin Kim, Namgyu Ho, James Thorne, and Se young Yun. 2023. Hare: Explainable hate speech detection with step-by-step reasoning. *Preprint*, arXiv:2311.00321.

Wajdi Zaghouani and Md Rafiul Biswas. 2025a. An annotated corpus of arabic tweets for hate speech analysis. *arXiv preprint arXiv:2505.11969*.

Wajdi Zaghouani and Md Rafiul Biswas. 2025b. Emohopespeech: An annotated dataset of emotions and hope speech in english and arabic. *arXiv preprint arXiv:2505.11959*.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgios Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055.

AraNLP at MAHED 2025 Shared Task: Using AraBERT for Text-based Hate and Hope Speech Classification

Enas A. Hakim Khalil
Systems & Information Department,
National Research Center (NRC),
Giza, Egypt
ea.khalil@nrc.sci.eg

Wafaa S. El-Kassas
Faculty of Computers and Information
Technology, The National Egyptian E-Learning
University (EELU), Giza, Egypt
wafaa.elkassas@gmail.com

The AraNLP system is designed for the MAHED Shared Task 2025 on text-based hate and hope classification in Arabic. The challenge was divided into three subtasks: (1) Text-based Hate and Hope Speech Classification, (2) Emotion, Offensive, and Directed Hate Detection (Multitask), and (3) Multimodal Hateful Meme Detection. The AraNLP system based on the AraBERT model achieved Macro F1-score 65% for Sub-task 1 and the results are published in the leaderboard, with rank 20. After submitting the results, the proposed model was updated to improve its performance and achieved Macro F1-score 70%, this makes the AraNLP system nearly equivalent to rank 4 in the leaderboard.

1 Introduction

Nowadays, social media platforms (e.g. Twitter, Facebook, etc.) facilitate expression of free speech. These platforms are very popular for users to have discussions and conversations and express their thoughts and opinions, but users sometimes use them to provide hate towards each other (Khezzar, Moursi, and Al Aghbari 2023). Moreover, anonymity provided to users on these platforms allows the spread of hate speech and other offensive material (Alwateer et al. 2025). Early and accurate detection of hate speech is important for maintaining a respectful and safe online environment, especially with the content's continuous and rapid growth which can lead to negative reactions from users (Al-Saqqa 2024). Therefore, there is an essential need to automatically detect and report occurrence of hate speech in text for different languages. In recent years, researchers focus on analyzing data shared on social media platforms but their attention is mainly directed towards the content written in English (Fat'hAlalim et al. 2025). In contrast, other languages such as Arabic needs more attention from researchers and needs more resources that facilitate the research tasks to get better results. Focusing on Arabic NLP tasks brings many challenges such as the lack of Arabic language resources, difficult grammatical structure, dialectal variations, and human annotation errors (Abdelsamie, Azab, and Hefny 2026). The goal of the MAHED 2025 Text-based Hate and Hope Speech Classification (Sub-task 1) is to classify Arabic text as hate, hope, or not_applicable. Such challenge is very important to encourage the research community to focus more on the tasks related to the Arabic content. The proposed AraNLP system uses the AraBERT model (Antoun, Baly, and Hajj 2020) and the experimental results demonstrate that the model achieves promising results for Sub-task 1.

2 Related Work

Hate speech is a very challenging and complex task especially with Arabic dialects as previous studies have often used multiple Arabic dialects combined in a single dataset without identifying the dialects used, which is challenging because it can lead to misidentification of non-hateful and hateful contexts related to a particular dialect (Abdelsamie et al. 2026). Moreover, the lack of adequate research on Arabic dialects and the lack of large, publicly available datasets highlight the need for more investigations about the Arabic hate speech detection (Fat'hAlalim et al. 2025).

Many studies about the detection of hate speech in Arabic tweets or social media posts in general have used different methods such as machine learning techniques, deep learning, the application of transfer learning, Arabic BERT-based models, and the Large Language Models (LLMs) (Al-Saqqa, Awajan, and Hammo 2024). In addition, researchers have examined hybrid models that combine different approaches to propose ensemble methods that incorporate multiple deep learning techniques to improve results (Al-Saqqa et al. 2024).

Proceedings of The Third Arabic Natural Language Processing Conference, pages 706–711
November 8-9, 2025 ©2025 Association for Computational Linguistics

The arHateDetector Framework was proposed to detect hate speech in the Arabic tweets by (Khezzar et al. 2023). The authors conducted several experiments to evaluate machine learning algorithms like logistic regression, Linear SVC, and Random Forest, in addition to deep learning models like AraBERT and Convolutional Neural Networks (CNNs). The experiments prove that AraBERT outperformed the other models achieving the best performance across seven different datasets.

An interpretable framework to detect hate speech in Arabic was developed based on LLMs by (Alwateer et al. 2025). The authors focus to enhance understanding of model decisions by combining interpretable machine learning methods with advanced Natural Language Processing (NLP) techniques in their proposed method. The results show the effectiveness of combining LLM with interpretability to provide a transparent solution for automated detection of harmful content.

In (Fat'hAlalim et al. 2025), the authors analyze Arabic hate speech detection using advanced transformer-based models across three datasets collected from different social media platforms. The analysis includes the effects of data augmentation, oversampling, and model interpretability using the LIME method. The monolingual transformer-based models achieved significant performance improvements. Besides, they applied cross-validation across datasets to evaluate the generalization capabilities of models.

In (Abdelsamie et al. 2026), the authors focused on understanding of dialect-specific hate speech and proposed a multi-task learning approach built upon transformer architecture to bridge the gap in hate speech detection across Arabic dialects. They used publicly available datasets from various dialects, the proposed model was designed to identify and differentiate subtle hate speech patterns and use shared representation knowledge across five Arabic dialects: Egyptian, Gulf, Saudi, Levant, and Algerian.

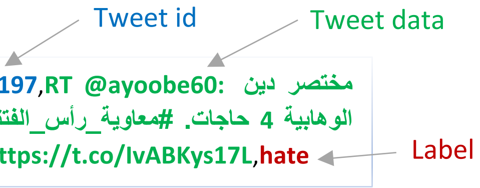
While deep learning, transfer learning, and ensemble learning approaches have shown potential, many challenges persist, specifically with Arabic language difficulties and dialectal variations (Fat'hAlalim et al. 2025). In (Ramos et al. 2024), the authors highlights that Transformer models consistently outperform other methods, but their high computational requirements suggest that hybrid approaches, combining deep learning with traditional machine learning, may be more suitable in certain contexts.

Although significant steps have been made in addressing low-resource languages like Arabic, there is still a need for further research work to improve inclusivity across a wider range of cultural and linguistic contexts (Ramos et al. 2024).

3 Data

The dataset (Zaghouani, et al., 2024) (Biswas & Zaghouani, 2025) (Biswas & Zaghouani, 2025) used in the MAHED 2025 Sub-task 1 consists of Arabic text (MSA and dialect) with train file of 6893 tweets, validation file of 1476 tweets, and test file of 1477 tweets. Figure 1 shows the detailed data format consisting of: tweet id, data and label. Each tweet is classified with one of three labels: hate, hope, or not_applicable.

Table 1 illustrates examples of classified tweets for different labels. The statistics of the dataset in terms of the number of tweets per label is provided in Table 2. It can be observed that the dataset is imbalanced in the count of tweets of the represented labels and this note explains later why the trained model is biased towards the not_applicable label.



7197,RT @ayoobe60: مختصر دين
الوهابية 4 حاجات. #معاوية_رأس_الفتنة
https://t.co/IvABKys17L,hate

Figure 1: Train data format.

Tweet	Label
كل المهاجرين لصوص ومجرمون يجب طردهم فوراً	Hate
معاً يمكننا بناء مستقبل أفضل لأطفالنا	Hope
اليوم هو يوم مشمس وجميل	not_applicable

Table 1: Examples of different classes of labeled data.

Label	Train	Validation	Test
hope	1892	409	422
hate	1304	262	287
not_applicable	3697	806	768
(All Labels)	6893	1476	1477

Table 2: Statistics of the dataset.

4 Methodology

The proposed system mainly uses the AraBERT model (Antoun et al. 2020). AraBERT uses the BERT-base configuration that has 12 encoder blocks, 768 hidden dimensions, 12 attention heads, 512 maximum sequence length, and a total of 110M parameters. AraBERT has different versions, and all models are available in the HuggungFace model page under the aubmind name. In the proposed system, the twitter AraBERT: 'aubmindlab/bert-base-arabertv02-twitter' model is used (HuggingFace n.d.). This model is selected because it was pretrained on 60M Arabic tweets with different dialects and Modern Standard Arabic (MSA) (i.e. it is a general Arabic or a simplified version that does not use diacritics and it is usually used in newspapers, tweets, etc.) which is similar in nature to the MAHED 2025 Sub-task 1 dataset. Besides, Arabertv02-twitter has better vocabulary coverage for slang, hashtags, and emojis.

Figure 2 shows the proposed system based on AraBERT model. The AraBERT Preprocessor (ArabertPreprocessor) is used for the training and validation files to clean the Arabic text. This step is important for handling the unique characteristics of the Arabic language, such as diacritics and ligatures. Then, the data is tokenized using the AraBERT tokenizer (AutoTokenizer) to convert the text into numerical tokens. After that, the distribution of sentence lengths is analyzed to help determine an appropriate maximum length. The maximum sentence length is determined to be 128 and truncate longer sentences to avoid performance degradation. 27, and 5 tweets of all the training, and validation tweets respectively exceed the maximum length and have been truncated.

Figure 2 shows the **text classification model** using the fine-tuning approach on a pre-trained **AraBERT model** (aubmindlab/bert-base-arabertv02-twitter). The training is processed with the training parameters such as learning rate, batch size, etc. Table 3 illustrates the used uniform hyper parameter settings for AraBERTv02-twitter.

Once the training is complete, the final fine-tuned model is saved to be used later in the prediction phase. The last step is to predict the output labels of new and unseen text data in the test set using the saved model with predict_labels and classification report libraries.

All experiments are carried out on the Google Colab environment and covers the entire machine learning pipeline from data preparation to model training and evaluation. The Google Colab platform is used with a NVIDIA L4 GPU, System RAM 6.6 / 53.0 GB, GPU RAM 1.3 / 22.5 GB, and Disk 40.7 / 235.7 GB.

Parameter	**Value**
adam_epsilon	1e-8
learning_rate	2 e -5
Number of train epochs	2
warmup_ratio	0
per_device_train_batch_size	16
per_device_eval_batch_size	128
gradient_accumulation_steps	2
do_eval	True
load_best_model_at_end	True
metric_for_best_model	'macro_f1'
greater_is_better	True
Seed	25

Table 3: Hyper parameters for AraBERTv02-twitter.

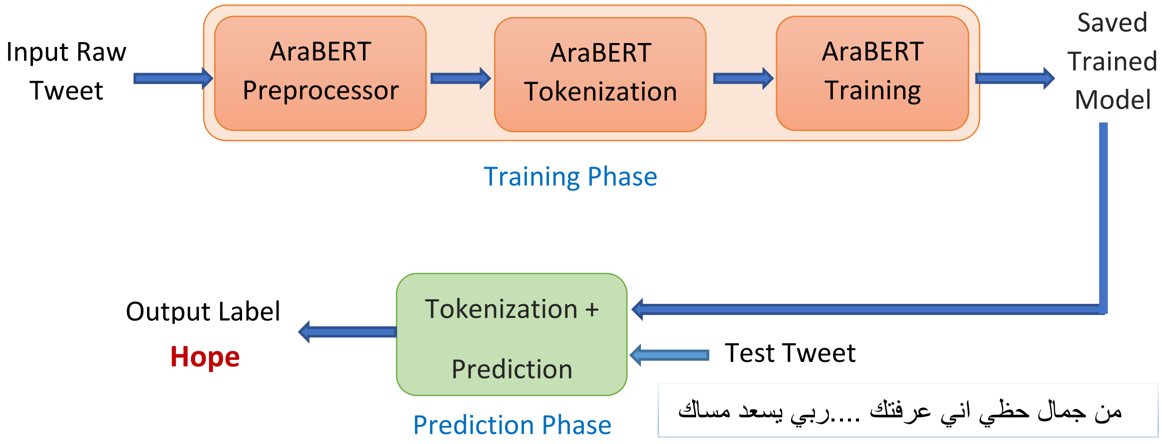


Figure 2: AraBERT Tweet Classification Model with labels hope, hate, or not applicable.

Epoch	Training Loss	Validation Loss	Macro F1	Accuracy	Macro Precision	Macro Recall
1	No log	0.667	0.650	0.680	0.668	0.637
2	No log	0.670	0.659	0.675	0.656	0.662

Table 4: Training Log Table.

5 Results

The performance of proposed system is evaluated on the validation set after each training epoch. The used evaluation metrics include **accuracy**, **macro F1-score**, **precision**, and **recall**, along with a **confusion matrix**.

The training results are recorded in table 4. Although the validation Loss (Model error on validation data) is slightly increased from 0.667 to 0.670 through the two epochs, which might indicate no improvement or slight overfitting but Macro F1 went from 0.650 to 0.659, which means the model got a little better. The Accuracy is dropped slightly from 0.680 to 0.675, also do Macro Precision from 0.668 to 0.656 meanwhile Macro Recall improved from 0.637 to 0.662.

The confusion matrix for training process shown in figure 3, the Class 0 (not_applicable) has approximately 19% errors, mostly confused with class 1 (hope) while class 1 got about 43% errors, mostly confused with class 0 and class 2 (hate) got 48% errors mostly confused with class 0. The model heavily counts on toward predicting class 0 when unsure.

To test the model, several experiments have been done. So, to differentiate between these experiments, they are referenced in this paper as Test 1, Test 2, and Test 3. Test 1 is the first experiment, and its results are published through competition in the leaderboard. Test 2 provides improved results than Test 1. The difference in results for the two tests comes from the predict_label library that was used in Test 2 instead of the **pipeline** in Test 1 for getting output labels. Test 2 is more accurate and represents better results.

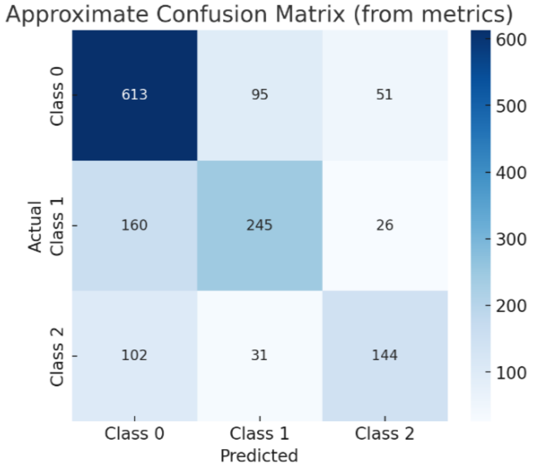


Figure 3: Confusion matrix for regular training process.

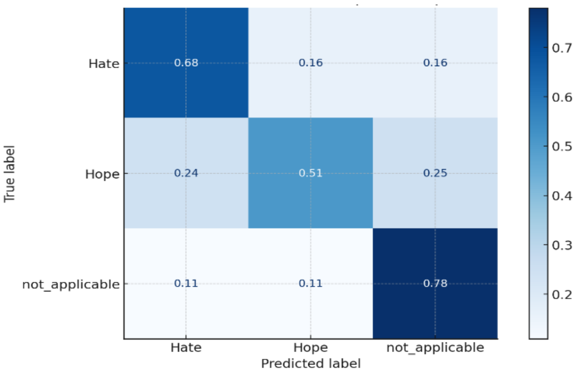


Figure 4: Normalized Confusion Matrix for Test 2.

From the classification report for test data (Test 2 results) in Table 5, it can be observed that while 68% of real hate is correctly predicted only 51% of real hope tweets are correctly found. 78% of not_applicable tweets are correctly predicted. The corresponding confusion matrix in figure 4 shows that about half of hope actual class tweets are confused with other labels, which confirms with the low recall results for hope class. These results suggest that the model generalize well because no performance drop from validation to test which suggests minimal overfitting.

In the third experiment (Test 3), both train and validation data were added in a single file called trainall.txt and this file was used to train the model with 5 fold Cross Validation and the number of epochs was increased from 2 (default value) to 5.

	Precision	Recall	F1-score	Support
Hate	0.70	0.68	0.69	287
Hope	0.69	0.51	0.59	422
not_applicable	0.68	0.78	0.72	768
Accuracy			0.68	1477
Macro Avg	0.69	0.66	0.67	1477
Weighted Avg	0.68	0.68	0.68	1477

Table 5: Classification Report for Test 2 (on Test Set).

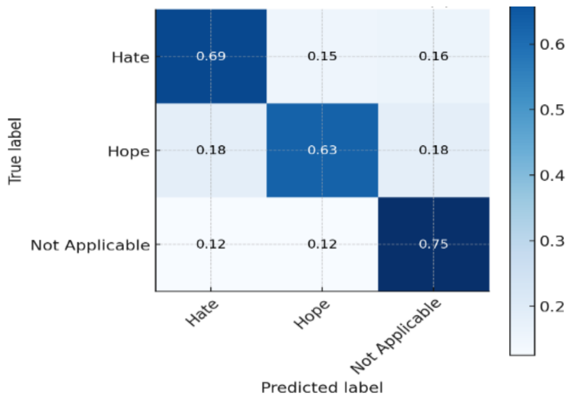


Figure 5: Normalized Confusion Matrix for Test 3.

These modifications improve the Test 3 results over all metrics compared to Test 1 and Test 2.

From the classification report for Test 3 (the **last improved results**) in Table 6, it can be observed that while 69% of real hate tweets is correctly predicted and only 63% of real hope tweets are correctly found. 75% of not_applicable tweets are correctly predicted. The corresponding confusion matrix for Test 3 in Figure 5 shows that the best class not_applicable (recall = 0.75) and the weakest one is hope class. The results suggest that the model generalize well because no performance drop from validation to test which suggests minimal overfitting.

Table 7 and Figure 6 compare the validation results for the training process with the predicted output labels for the test data using the saved trained model. The three experiments results for test data are compared in Table 7. The difference between the three experiments for test data have been explained earlier in this section.

In the three test experiments, the model generalizes well with no performance drop from Validation to Test, which suggests minimal overfitting. It is clear that augmenting more data samples in training helps to climb higher in performance in the experiment Test 3 which achieves the better results compared to Test 1 and Test 2 results.

	Precision	**Recall**	**F1-score**	**Support**
Hate	0.70	0.69	0.69	287
Hope	0.70	0.63	0.66	422
not_applicable	0.71	0.75	0.7	768
Accuracy			0.70	1477
Macro Avg	0.70	0.69	0.70	1477
Weighted Avg	0.70	0.69	0.70	1477

Table 6: Classification Report for Test 3 (on Test Set).

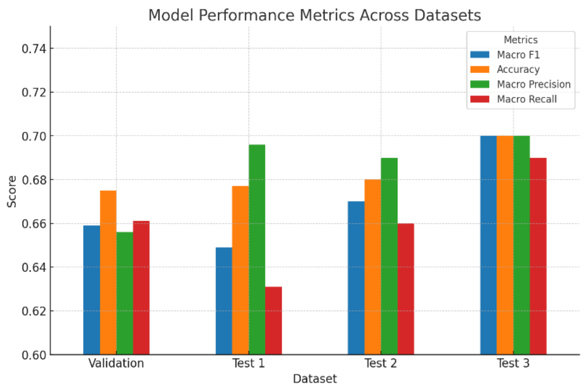


Figure 6: Validation vs. Test Results Comparison.

	Macro F1	**Accuracy**	**Macro Precision**	**Macro Recall**
Validation	0.659	0.675	0.6561	0.661
Test 1 (Leaderboard)	0.649	0.677	0.696	0.631
Test 2	0.67	0.68	0.69	0.66
Test 3	**0.70**	**0.70**	**0.70**	**0.69**

Table 7: Results of Validation and Test data**.**

6 Conclusion

Recently, the detection of hate speech from social media such as Twitter gains attention of researchers. Real-time detection of harmful content is essential to safeguarding vulnerable communities, so it becomes essential to make continued research and development in Arabic hate speech detection.

This paper focuses on the domain of detecting hate and hope speech in Arabic. AraBERT model is used to detect hate and hope speech in Arabic Tweets. Three different experiments have been done on the test data and the results are compared and explained. The evaluation of AraNLP system shows promising and better results in Test 3 than Test 1 and Test 2.

Future work includes evaluating the proposed system on various hate speech datasets to evaluate the performance of both the multilingual and monolingual models. Also, oversampling techniques can be used to address the class imbalance to improve the proposed model performance. In addition, conducting extensive experiments by using and evaluating different transformer-based models and Large Language Models (LLMs) to achieve better results.

References

Abdelsamie, Mahmoud Mohamed, Shahira Shaaban Azab, and Hesham A. Hefny. 2026. “The Dialects Gap: A Multi-Task Learning Approach for Enhancing Hate Speech Detection in Arabic Dialects.” *Expert Systems with Applications* 295:128584. doi:https://doi.org/10.1016/j.eswa.2025.128584.

Al-Saqqa, Samar. 2024. “Hate Speech Detection of Arabic Text Using Deep Learning and Transfer Learning Models.” PhD Thesis, Princess Sumaya University for Technology (Jordan).

Al-Saqqa, Samar, Arafat Awajan, and Bassam Hammo. 2024. “A Survey of Hate Speech Detection for Arabic Social Media: Methods and Datasets.” *Procedia Computer Science* 251:224–31. doi:https://doi.org/10.1016/j.procs.2024.11.104.

Alwateer, M., I. Gad, M. Elmarhomy, G. Elmarhomy, H. Hashim, M. Almaliki, and E. S. Atlam. 2025. “Interpretable Arabic Hate Speech Detection Using Large Language Model.” Pp. 1–8 in *2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC)*.

Antoun, Wissam, Fady Baly, and Hazem Hajj. 2020. “AraBERT: Transformer-Based Model for Arabic Language Understanding.” Pp. 9–15 in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, edited by H. Al-Khalifa, W. Magdy, K. Darwish, T. Elsayed, and H. Mubarak. Marseille, France: European Language Resource Association.

Biswas, Md. Rafiul, and Wajdi Zaghouani. 2025a. “An Annotated Corpus of Arabic Tweets for Hate Speech Analysis.” *CoRR*. https://arxiv.org/abs/2505.11969.

Biswas, Md. Rafiul, and Wajdi Zaghouani. 2025b. “EmoHopeSpeech: An Annotated Dataset of Emotions and Hope Speech in English and Arabic.” *CoRR* abs/2505.11959. https://arxiv.org/abs/2505.11959.

Fat’hAlalim, Ahmed, Yongjian Liu, Qing Xie, and Nahla Ibrahim. 2025. “Advancements in Transformer-Based Models for Enhanced Hate Speech Detection in Arabic: Addressing Dialectal Variations and Cross-Platform Challenges.” *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 24(8). doi:10.1145/3748492.

HuggingFace. n.d. “Bert-Base-Arabertv02-Twitter.” Retrieved September 12, 2025. https://huggingface.co/aubmindlab/bert-base-arabertv02-twitter.

Khezzar, Ramzi, Abdelrahman Moursi, and Zaher Al Aghbari. 2023. “ArHateDetector: Detection of Hate Speech from Standard and Dialectal Arabic Tweets.” *Discover Internet of Things* 3(1):1. doi:10.1007/s43926-023-00030-9.

Ramos, Gil, Fernando Batista, Ricardo Ribeiro, Pedro Fialho, Sérgio Moro, António Fonseca, Rita Guerra, Paula Carvalho, Catarina Marques, and Cláudia Silva. 2024. “A Comprehensive Review on Automatic Hate Speech Detection in the Age of the Transformer.” *Social Network Analysis and Mining* 14(1):204. doi:10.1007/s13278-024-01361-3.

Zaghouani, Wajdi, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgio Mikros, Abul Hasnat, and Firoj Alam. 2025. “MAHED Shared Task: Multimodal Detection of Hope and Hate Emotions in Arabic Content.” in *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*. Suzhou, China: Association for Computational Linguistics.

Zaghouani, Wajdi, Hamdy Mubarak, and Md. Rafiul Biswas. 2024. “So Hateful! Building a Multi-Label Hate Speech Annotated Arabic Dataset.” Pp. 15044–55 in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Torino, Italia: ELRA and ICCL.

Thinking Nodes at MAHED: A Comparative Study of Multimodal Architectures for Arabic Hateful Meme Detection

Itbaan Safwan
Institute of Business Administration
Karachi, Pakistan
i.safwan.26197@khi.iba.edu.pk

Abstract

This paper describes our system for Task 3 of the Arabic NLP 2025 competition: detecting hateful content in Arabic memes. The task requires a robust understanding of both visual and textual information and their interplay. We developed and compared three distinct multimodal fusion architectures: a Cross-Attention model, a progressive CNN-based fusion model, and a two-stage model using custom-trained embeddings with a gated fusion classifier. All models leverage pre-trained CLIP and MARBERT encoders for image and text representation, respectively. We detail our approach to handling the significant class imbalance in the dataset through data re-splitting and the application of a weighted Focal Loss. Our post-competition analysis, training on all available data, shows that the CNN-based fusion model achieved the highest macro F1-score of 0.779, demonstrating the effectiveness of its hierarchical feature extraction for this task.

1 Introduction

The proliferation of memes on social media has transformed them into a potent medium for communication, but also for the spread of hate speech. Detecting hateful content within memes is a challenging multimodal task, as the malicious intent often arises not from the image or text in isolation, but from their complex and often ironic interplay. This paper presents our contribution to the Arabic NLP 2025 Shared Task 3 on Multimodal Hateful Meme Detection (Zaghouani et al., 2025), which focuses on classifying Arabic memes as hateful or not hateful.

Previous work has established benchmarks for multimodal hate speech detection, often focusing on English memes and exploring various fusion strategies (Kiela et al., 2021). While recent efforts have begun to build valuable resources for Arabic, such as the ArMeme dataset (Alam et al., 2024b), a systematic comparison of different deep fusion architectures specifically for hateful Arabic memes remains an area ripe for exploration. The optimal way to combine visual and textual cues—whether by capturing global context or local patterns—is not yet well understood for this specific domain.

To address this gap, we conduct a comparative analysis of three distinct fusion architectures, leveraging powerful pre-trained CLIP and MARBERT encoders as our backbones. We investigate a global Cross-Attention mechanism, a localized progressive CNN-based approach, and a two-stage Custom Embedding model. A key part of our methodology was also addressing the severe class imbalance in the dataset through stratified re-splitting and a weighted Focal Loss. Our experiments reveal that the progressive CNN model achieves the highest performance, demonstrating the effectiveness of learning hierarchical local features for this task.

The main contributions of this paper are as follows:

1. We provide a direct, empirical comparison of three different multimodal fusion strategies (Cross-Attention, CNN, and a two-stage contrastive approach) on the task of Arabic hateful meme detection.

2. We demonstrate an effective methodology for mitigating severe class imbalance through a combination of stratified data splitting and a weighted Focal Loss function.

3. Our post-competition analysis provides a strong performance benchmark, with our best model achieving a macro F1-score of 0.779 and highlighting the superiority of the CNN-based fusion approach for this specific task.

Our code is available at a public repository[1]

[1] https://github.com/itbaans/ArabicNLP-2025

Proceedings of The Third Arabic Natural Language Processing Conference, pages 712–719
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Related Work

Our research is situated at the intersection of multimodal machine learning, hate speech detection, and Arabic Natural Language Processing. This section reviews key advancements in these areas to contextualize our contributions.

2.1 Multimodal Hate Speech Detection

The task of identifying hate speech has expanded from text-only analysis to the more complex domain of multimodal content. The Hateful Memes Challenge by Kiela et al. (2021) was a seminal work that established a benchmark for the task, highlighting cases where models fail if they cannot reason jointly about the image and text. Early approaches often relied on simple fusion, such as concatenating features from separate unimodal encoders. More recent works have focused on developing sophisticated deep fusion mechanisms. Cross-attentional models, which learn to align and integrate features from different modalities, have shown strong performance in various vision-and-language tasks and have been widely adopted for meme analysis (Tan and Bansal, 2019). Our work contributes to this line of research by directly comparing a cross-attention architecture with alternative fusion strategies.

2.2 Arabic Multimodal and Hate Speech Resources

While multimodal research has historically been dominated by English-language resources, there has been a significant and growing effort to develop datasets and models for Arabic. For text-based hate speech, Zaghouani et al. (2024) provided a large, richly annotated dataset of Arabic tweets, demonstrating the effectiveness of transformer-based models like AraBERT for the task. The challenge of multimodality in Arabic memes has been tackled more recently. Alam et al. (2024b) introduced ArMeme, the first major dataset for multimodal analysis of Arabic memes, providing annotations for various tasks including propaganda detection. Building on this, Alam et al. (2024a) explored the critical intersection between propaganda and hate speech in memes, using a multi-agent LLM approach to annotate and analyze this relationship. Concurrently, efforts like the ArAIEval shared task have spurred research into multimodal propaganda detection, with participants such as Shah et al. (2024) successfully employing fusion architectures combining BERT with vision models like ConvNeXt.

Our work builds directly on these foundational efforts. While previous studies have focused on creating resources or detecting propaganda, our paper provides a focused, comparative study of different deep fusion architectures specifically for the nuanced task of hate speech detection in Arabic memes, using the dataset provided by the ArabicNLP 2025 shared task.

3 System Overview

To conduct our comparative analysis, we developed three distinct multimodal architectures. All models share a common foundation, utilizing powerful pre-trained encoders for initial feature representation, but differ significantly in their strategy for fusing these features. A detailed breakdown of each model's architecture, including layer configurations and hyperparameters, is available in Appendix A.

3.1 Backbone Encoders

For visual feature extraction, we employ the vision transformer from **openai/clip-vit-base-patch32** (Radford et al., 2021). For the corresponding Arabic captions, we use **UBC-NLP/MARBERT** (Abdul-Mageed et al., 2021). In our end-to-end models, we adopt a partial fine-tuning strategy, unfreezing only the final two layers of each encoder to adapt them to the specific domain of Arabic memes while preserving their rich, general-purpose knowledge.

3.2 Fusion Architectures

Model 1: Cross-Attention Fusion This model (Figure 1) is designed to capture the global, interdependent context between modalities. Inspired by co-attentional transformers (Tan and Bansal, 2019), it uses a bidirectional cross-attention mechanism where image and text features query each other to form contextually enriched representations before being pooled and classified.

Model 2: CNN-based Fusion In contrast, this architecture (Figure 2) aims to learn localized, compositional features. Motivated by the effectiveness of convolutions for fusing aligned sequences (Zadeh et al., 2017), this model uses a stack of 1D convolutional layers to progressively fuse the image and text embedding sequences, allowing it to build a hierarchical understanding of their interaction.

Model 3: Custom Embedding Fusion This model (Figure 3) follows a two-stage pipeline to decouple modality alignment from classification. In the first stage, we pre-train a custom dual-encoder model using a contrastive loss, following the CLIP methodology (Radford et al., 2021), to align the image and text features into a shared embedding space. In the second stage, a lightweight classifier fuses these pre-computed embeddings using a gated mechanism (Arevalo et al., 2017), which dynamically weights the contribution of each modality for the final prediction.

4 Experimental Setup

4.1 Dataset and Preprocessing

The original dataset, introduced by Zaghouani et al. (2024) and analyzed for multimodal hate speech by Alam et al. (2024a), was provided with separate train, development, and test splits. We observed a significant class imbalance, particularly in the development set, which could skew validation performance. To create a more stable training and evaluation environment, we combined all provided labeled data (train, dev, and the labeled test set from a previous phase) and performed a new stratified split, allocating 70% for training and 30% for validation. This ensured that the class proportions were consistent across both splits.

4.2 Handling Class Imbalance

The dataset is heavily skewed towards the ’not-hate’ class. To mitigate this, we employed a weighted Focal Loss (Lin et al., 2018) instead of standard cross-entropy. Focal Loss addresses class imbalance by down-weighting the loss assigned to well-classified examples, thereby focusing training on hard, misclassified examples. It is defined as:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \qquad (1)$$

We set the focusing parameter $\gamma = 2$. The balancing parameter α_t was set using class weights computed inversely proportional to class frequencies:

$$w_c = \frac{N}{2 \times N_c} \qquad (2)$$

where N is the total number of samples, and N_c is the number of samples in class c. These weights were passed to the loss function, increasing the penalty for misclassifying the minority ’hate’ class.

4.3 Implementation Details

All models were trained using the AdamW optimizer with a weight decay of 1×10^{-5}. For the end-to-end models (Cross-Attention, CNN), we used a learning rate of 2×10^{-5}. For the lightweight fusion classifier (Custom Embedding), we used a higher learning rate of 5×10^{-5}. All experiments were run with a batch size of 32. We used a ‘ReduceLROnPlateau‘ scheduler to decrease the learning rate if the validation F1-score did not improve for 2 epochs. Early stopping was implemented with a patience of 10-15 epochs to prevent overfitting.

5 Results and Analysis

We report two sets of results: pre-submission results based on models trained only on our 70% training split, and post-submission results where models were trained on the full combined dataset (train + validation) and evaluated on the official test set with gold labels. The official evaluation metric is macro F1-score.

5.1 Pre-Submission Results

For the official competition submission, we inadvertently trained our models only on our 70% training split, not the full available labeled data. The CNN and Cross-Attention models were submitted to the leaderboard. Due to time constraints, the Custom Embedding model was not submitted, but we report its projected score on the test set for comparison. Table 1 summarizes these findings.

The CNN model achieved the highest F1-score on our validation set, but both submitted models performed almost identically on the official test set. The Custom Embedding model, despite its lower validation score, shows a strong projected test score, indicating its potential.

5.2 Post-Submission Analysis

After the competition, the test set gold labels were released. This allowed us to conduct a more thorough analysis by training our models on all available labeled data (our 70% train + 30% validation splits combined) and evaluating on the official test set. The results are shown in Table 2.

Impact of Training Data Size A key finding is the significant performance boost observed across all models when trained on the full dataset versus the partial split. The CNN Fusion model’s F1-score, for instance, jumped from 0.718 to 0.779 (+6.1 points). This highlights that our models were

Model	Validation Set (Our Split)				Official Test Set			
	Val F1	Precision	Recall	Accuracy	Official F1	Precision	Recall	Accuracy
Cross-Attention	0.692	0.668	0.750	0.824	0.719	0.733	0.714	0.740
CNN Fusion	**0.727**	**0.696**	**0.802**	0.840	0.718	**0.776**	0.711	**0.754**
Custom Emb.	0.690	0.683	0.698	**0.853**	**0.720***	0.752	**0.713**	0.748

Table 1: Pre-submission results. Models were trained on a 70% split of the data. Metrics for the validation set are macro-averaged for F1, Precision, and Recall. Official Test F1 is from the CodaLab leaderboard or our projection based on gold labels (*).

Model	Test F1 (Full Data)
Cross-Attention	0.765
CNN Fusion	**0.779**
Custom Emb. Fusion	0.765

Table 2: Post-submission results. Models were trained on all available labeled data and evaluated on the official test set.

data-hungry and that leveraging all available annotations was critical for achieving optimal performance. Our pre-submission results were therefore limited by our experimental oversight.

Model Comparison In the post-submission setting, the CNN Fusion model emerged as the clear top performer. Its ability to extract and fuse localized features through convolutions appears to be more effective for this task than the global context mixing of cross-attention. The progressive nature of the fusion may also allow it to build more robust cross-modal representations. The Cross-Attention and Custom Embedding models achieved identical, strong scores, demonstrating their viability, but were ultimately outperformed by the CNN-based approach. The two-stage custom embedding approach is particularly noteworthy for its efficiency at inference time, as it only requires running a very small classifier once embeddings are pre-computed.

6 Conclusion

In this paper, we presented a comparative study of three distinct multimodal architectures—Cross-Attention, progressive CNN, and a two-stage Custom Embedding fusion—for the task of Arabic hateful meme detection. Our investigation confirmed that leveraging powerful pre-trained encoders like CLIP and MARBERT provides a strong foundation. Our findings underscore two critical aspects for this task: first, the necessity of robust techniques like weighted Focal Loss to handle severe class imbalance, and second, the significant impact of training data volume on final performance. Our post-submission analysis identified the progressive CNN-based fusion architecture as the most effective, achieving a final macro F1-score of 0.779 and suggesting that learning localized, hierarchical cross-modal interactions is a particularly robust strategy for this domain.

6.1 Limitations and Future Work

Despite these promising results, our study has several limitations. A primary concern is the models' propensity to overfit, evidenced by a decline in validation performance even as training loss decreased. This suggests that the complex architectures may have memorized spurious correlations from the relatively small dataset rather than learning generalizable features of hate speech. Another key limitation is the "black-box" nature of our fusion mechanisms, which hinders the interpretability required for reliable real-world moderation systems. Furthermore, our models do not explicitly process text embedded within images, a common feature in memes.

Future work should directly address these issues. A promising direction to mitigate both data scarcity and overfitting is to employ knowledge distillation (Hinton et al., 2015). One could leverage a powerful Vision-Language Model (VLM), such as those from the CLIP or BLIP families (Radford et al., 2021; Li et al., 2022), as a "teacher" to generate a large, pseudo-labeled dataset with soft probability distributions. A more compact "student" model, like our CNN architecture, could then be trained to mimic the teacher's nuanced outputs, transferring its reasoning capabilities into a more efficient and robust model. To improve interpretability, future research could focus on generating saliency maps to highlight which image regions and text tokens most influence a prediction, providing a clearer view into the model's decision-making process.

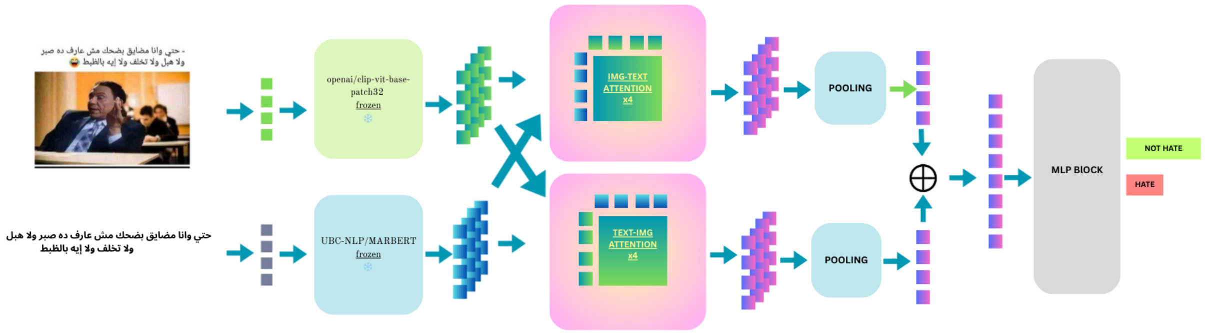

- حتي وانا مضايق بضحك مش عارف ده صبر
ولا هبل ولا تخلف ولا إيه بالظبط 😂
حتي وانا مضايق بضحك مش عارف ده صبر ولا هبل
ولا تخلف ولا إيه بالظبط
openai/clip-vit-base-patch32
frozen
UBC-NLP/MARBERT
frozen
IMG-TEXT ATTENTION x4
TEXT-IMG ATTENTION x4
POOLING
POOLING
MLP BlOCK
NOT HATE
HATE

Figure 1: Architecture of the Cross-Attention Fusion model.

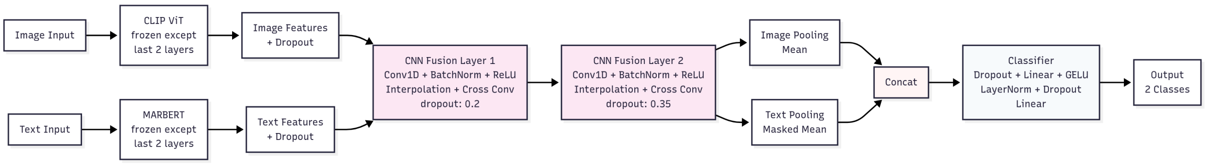


Figure 2: Architecture of the progressive CNN-based Fusion model.

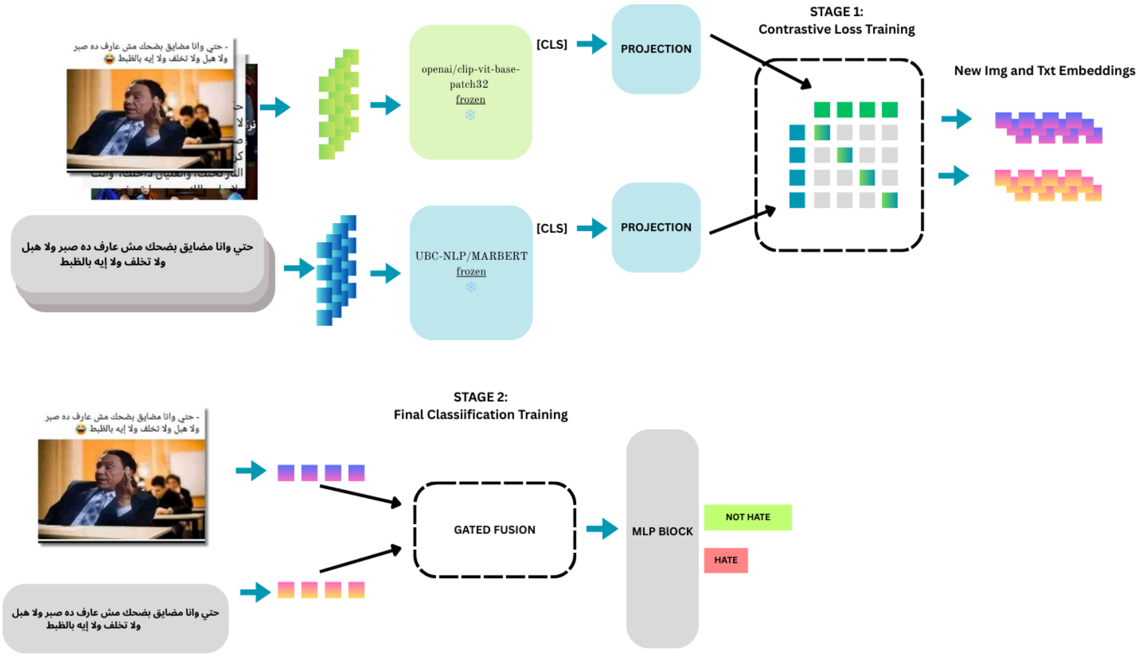

- حتي وانا مضايق بضحك مش عارف ده صبر
ولا هبل ولا تخلف ولا إيه بالظبط 😂
حتي وانا مضايق بضحك مش عارف ده صبر ولا هبل
ولا تخلف ولا إيه بالظبط
openai/clip-vit-base-patch32
frozen
UBC-NLP/MARBERT
frozen
[CLS]
[CLS]
PROJECTION
PROJECTION
STAGE 1:
Contrastive Loss Training
New Img and Txt Embeddings
STAGE 2:
Final Classiification Training
- حتي وانا مضايق بضحك مش عارف ده صبر
ولا هبل ولا تخلف ولا إيه بالظبط 😂
حتي وانا مضايق بضحك مش عارف ده صبر ولا هبل
ولا تخلف ولا إيه بالظبط
GATED FUSION
MLP BlOCK
NOT HATE
HATE

Figure 3: Architecture of the two-stage Custom Embedding Fusion model.

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.

Firoj Alam, Md Rafiul Biswas, Uzair Shah, Wajdi Zaghouani, and Georgios Mikros. 2024a. Propaganda to hate: A multimodal analysis of arabic memes with multi-agent llms. In *International Conference on Web Information Systems Engineering*, pages 380–390. Springer.

Firoj Alam, Abul Hasnat, Fatema Ahmad, Md. Arid Hasan, and Maram Hasanain. 2024b. ArMeme: Propagandistic content in Arabic memes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21071–21090, Miami, Florida, USA. Association for Computational Linguistics.

John Arevalo, Thamar Solorio, Manuel Montes y Gómez, and Fabio A. González. 2017. Gated multimodal units for information fusion. *Preprint*, arXiv:1702.01992.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2021. The hateful memes challenge: Detecting hate speech in multimodal memes. *Preprint*, arXiv:2005.04790.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pretraining for unified vision-language understanding and generation. *Preprint*, arXiv:2201.12086.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *Preprint*, arXiv:1708.02002.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Uzair Shah, Md. Rafiul Biswas, Marco Agus, Mowafa Househ, and Wajdi Zaghouani. 2024. MemeMind at ArAIEval shared task: Generative augmentation and feature fusion for multimodal propaganda detection in Arabic memes through advanced language and vision models. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 467–472, Bangkok, Thailand. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *Preprint*, arXiv:1908.07490.

Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor fusion network for multimodal sentiment analysis. *Preprint*, arXiv:1707.07250.

Wajdi Zaghouani, Md Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, Georgio Mikros, Abul Hasnat, and Firoj Alam. 2025. MAHED shared task: Multimodal detection of hope and hate emotions in arabic content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Wajdi Zaghouani, Hamdy Mubarak, and Md. Rafiul Biswas. 2024. So hateful! building a multi-label hate speech annotated Arabic dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15044–15055, Torino, Italia. ELRA and ICCL.

A Model Architectures and Implementation Details

A.1 CNN-Based Multimodal Fusion Model

The CNN-based fusion model (CNNMultiModalModel) employs 1D convolutional layers to process and fuse multimodal embeddings from CLIP-ViT and MARBERT encoders.

A.1.1 CNNFusionLayer Components

The core fusion component uses 1D convolutions for cross-modal interaction:

- **Input Processing**: Separate 1D convolutions for image and text embeddings with kernel size 3
- **Cross-Modal Fusion**: Concatenation followed by 1×1 convolution for dimensionality reduction
- **Normalization**: BatchNorm1d without affine parameters to prevent overfitting
- **Regularization**: Progressive dropout rates (0.2 + layer_index × 0.15)

A.1.2 Backbone Configuration

- **Vision Encoder**: CLIP-ViT-Base-Patch32 (768-dimensional embeddings)
- **Text Encoder**: MARBERT (768-dimensional embeddings)
- **Selective Unfreezing**: Only the last 2 layers of each encoder are trainable
- **Regularization**: 0.3 dropout applied to backbone outputs

A.1.3 Classification Head

The final classification component consists of:

$$\begin{aligned}\text{Classifier} = \text{Sequential}(&\\ &\text{Dropout}(0.5),\\ &\text{Linear}(\text{final\_dim} \times 2, \text{final\_dim}),\\ &\text{GELU}(),\\ &\text{LayerNorm}(\text{final\_dim}),\\ &\text{Dropout}(0.4),\\ &\text{Linear}(\text{final\_dim}, 2))\end{aligned}$$

A.2 Cross-Attention Fusion Model

The Advanced Fusion Model (AdvancedFusionModel) utilizes multi-head cross-attention mechanisms to enable bidirectional information exchange between visual and textual modalities.

A.2.1 CrossAttentionFusion Module

The fusion mechanism implements bidirectional cross-attention:

- **Text-to-Image Attention**: $\text{Att}_{t2i} = \text{MultiHeadAttn}(Q = I, K = T, V = T)$
- **Image-to-Text Attention**: $\text{Att}_{i2t} = \text{MultiHeadAttn}(Q = T, K = I, V = I)$
- **Pooling Strategies**: Support for mean, max, and learnable attention pooling
- **Feature Concatenation**: Final fusion via concatenation of pooled representations

A.2.2 Attention Pooling Mechanism

For attention-based pooling, learnable query vectors are employed:

$$\text{pooled\_img} = \text{Attention}(Q = q_{\text{img}}, K = \text{Att}_{t2i}, V = \text{Att}_{t2i}) \tag{3}$$

$$\text{pooled\_txt} = \text{Attention}(Q = q_{\text{txt}}, K = \text{Att}_{i2t}, V = \text{Att}_{i2t}) \tag{4}$$

where q_{img} and q_{txt} are randomly initialized learnable parameters.

A.2.3 Model Configuration

- **Attention Heads**: 4 heads for cross-attention modules
- **Frozen Backbones**: Complete freezing of CLIP-ViT and MARBERT parameters
- **Projection Layer**: 512-dimensional intermediate representation
- **Dropout Rates**: 0.4 for projection layer, 0.2 for classification head

A.3 Custom CLIP-Arabic with Embeddings Fusion

The custom approach involves pre-training a CLIP-style model on Arabic multimodal data, followed by embedding-based classification using various fusion strategies.

A.3.1 CLIPArabic Pre-training

The custom CLIP model implements contrastive learning:

- **Image Encoder**: Frozen CLIP-ViT-Base-Patch32
- **Text Encoder**: Frozen MARBERT

- **Projection Heads**: Linear layers mapping to 512-dimensional space
- **Contrastive Loss**: Symmetric cross-entropy on image-text similarity matrix

The contrastive loss function is defined as:

$$\mathcal{L}_{\text{contrastive}} = \frac{1}{2}(\mathcal{L}_{i2t} + \mathcal{L}_{t2i}) \quad (5)$$

$$\text{where} \quad \mathcal{L}_{i2t} = \text{CrossEntropy}(\tau \cdot \mathbf{I}\mathbf{T}^T, \mathbf{y}) \quad (6)$$

$$\mathcal{L}_{t2i} = \text{CrossEntropy}(\tau \cdot \mathbf{T}\mathbf{I}^T, \mathbf{y}) \quad (7)$$

with τ being the learnable temperature parameter and $\mathbf{y}$ the identity matrix labels.

A.3.2 Embeddings-Based Classification

The PrecomputedEmbeddingsClassifier supports multiple fusion strategies:

Gated Fusion (Best Performing):

$$\begin{aligned} g_{\text{img}} &= \sigma(W_{g,i}\mathbf{e}_{\text{img}} + b_{g,i}) \\ g_{\text{txt}} &= \sigma(W_{g,t}\mathbf{e}_{\text{txt}} + b_{g,t}) \\ \mathbf{h}_{\text{fused}} &= g_{\text{img}} \odot \text{ReLU}(W_i\mathbf{e}_{\text{img}}) \\ &\quad + g_{\text{txt}} \odot \text{ReLU}(W_t\mathbf{e}_{\text{txt}}) \end{aligned}$$

Alternative Fusion Methods:

- **Concatenation**: $\mathbf{h}_{\text{fused}} = [\mathbf{e}_{\text{img}}; \mathbf{e}_{\text{txt}}]$
- **Element-wise Addition**: $\mathbf{h}_{\text{fused}} = W_i\mathbf{e}_{\text{img}} + W_t\mathbf{e}_{\text{txt}}$
- **Element-wise Multiplication**: $\mathbf{h}_{\text{fused}} = W_i\mathbf{e}_{\text{img}} \odot W_t\mathbf{e}_{\text{txt}}$

A.4 Training Configuration and Hyperparameters

Table 3: Training hyperparameters for all models

Parameter	CNN	Cross-Attn	Custom CLIP
Learning Rate	$2{\times}10^{-5}$	$2{\times}10^{-5}$	$5{\times}10^{-5}$
Batch Size	32	32	32
Max Epochs	30	30	30
Early Stop Patience	10	10	10
Weight Decay	$1{\times}10^{-5}$	$1{\times}10^{-5}$	$1{\times}10^{-5}$
Gradient Clipping	1.0	1.0	1.0
Loss Function	Focal	Focal	Focal
Scheduler	ReduceLR	ReduceLR	ReduceLR

A.4.1 Focal Loss Configuration

All models employ Focal Loss to address class imbalance:

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (8)$$

where $\gamma = 2.0$ and α_t are computed based on inverse class frequencies.

A.5 Model Training Curves

To further illustrate the overfitting behavior discussed in the Limitations section, Figure 4 shows the training loss and test macro F1-score progression for all three models. In each case, the test F1-score (solid lines) peaks relatively early in training, after which it either stagnates or degrades, even as the training loss (dashed lines) continues to decrease. This divergence is a clear indicator that the models began to memorize the training data rather than learning generalizable patterns.

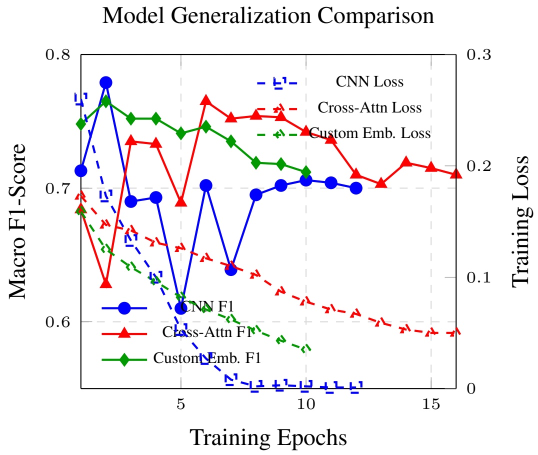


Figure 4: Comparison of training loss (dashed lines, right axis) vs. test macro F1-score (solid lines, left axis) for all models.

NADI 2025:
The First Multidialectal Arabic Speech Processing Shared Task

Bashar Talafha[λ] **Hawau Olamide Toyin**[ξ] **Peter Sullivan**[λ] **AbdelRahim Elmadany**[λ]
Abdurrahman Juma[γ] **Amirbek Djanibekov**[ξ] **Chiyu Zhang**[λ] **Hamad Alshehhi**[ξ]
Hanan Aldarmaki[ξ] **Mustafa Jarrar**[αγ] **Nizar Habash**[ηξ] **Muhammad Abdul-Mageed**[λ]
[λ]The University of British Columbia [α]Hamad Bin Khalifa University
[γ]Birzeit University [ξ]MBZUAI [η]NYU Abu Dhabi
{btalafha@mail.,a.elmadany@,muhammad.mageed@}ubc.ca

Abstract

We present the findings of the sixth Nuanced Arabic Dialect Identification (NADI 2025) Shared Task, which focused on Arabic speech dialect processing across three subtasks: spoken dialect identification (Subtask 1), speech recognition (Subtask 2), and diacritic restoration for spoken dialects (Subtask 3). A total of 44 teams registered, and during the testing phase, 100 valid submissions were received from eight unique teams. The distribution was as follows: 34 submissions for Subtask 1 "*five teams*", 47 submissions for Subtask 2 "*six teams*", and 19 submissions for Subtask 3 "*two teams*". The best-performing systems achieved 79.8% accuracy on Subtask 1, 35.68/12.20 WER/CER (overall average) on Subtask 2, and 55/13 WER/CER on Subtask 3. These results highlight the ongoing challenges of Arabic dialect speech processing, particularly in dialect identification, recognition, and diacritic restoration. We also summarize the methods adopted by participating teams and briefly outline directions for future editions of NADI.[1]

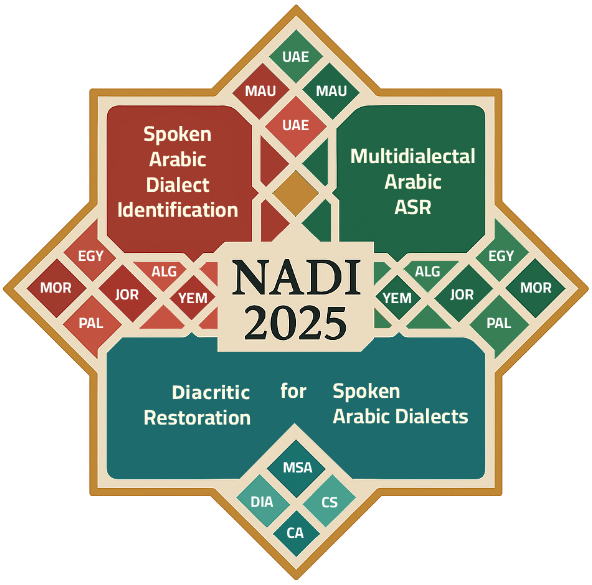

UAE
MAU
MAU
UAE
Spoken Arabic Dialect Identification
Multidialectal Arabic ASR
EGY
MOR
JOR
ALG
YEM
PAL
NADI 2025
ALG
YEM
JOR
EGY
MOR
PAL
Diacritic Restoration for Spoken Arabic Dialects
MSA
DIA
CS
CA

Figure 1: Overview of the **NADI 2025** shared tasks.

1 Introduction

Spoken Arabic exhibits a remarkable degree of linguistic diversity. Beyond Modern Standard Arabic (MSA) and Classical Arabic (CA), which have historically dominated computational work, Arabic encompasses numerous regional and national dialects that differ across all linguistic levels (phonology, morphology, lexicon, and syntax) and in discourse/pragmatics (Talafha et al., 2024; Jarrar et al., 2023). These varieties also frequently exhibit intra- and inter-sentential code-switching with other languages (Abdul-Mageed et al., 2024). These varieties dominate everyday communication across the Arab world yet remain under-represented in annotated datasets and resources (Bouamor et al., 2018; Darwish et al., 2021; Abdul-Mageed et al., 2020, 2023). At the same time, many downstream applications—from automatic transcription and virtual assistants to text-to-speech and educational tools—depend on accurate handling of dialectal speech and the diacritics that indicate short vowels and phonological features. Existing systems trained on CA/MSA (Elmadany et al., 2023a; Toyin et al., 2023) often ignore these diacritics or assume text forms, leaving a large gap between technology and real-world usage.

NADI shared task series, hosted at the Arabic-NLP conference[2] since 2020, was created to alleviate this bottleneck by providing curated datasets and standardized evaluation settings for dialect identification, translation and related tasks (Abdul-Mageed et al., 2020, 2021; Abdul-Mageed et al., 2022, 2023, 2024). These earlier, text-focused editions—together with the general observation that

[1] The official leaderboards and datasets for NADI 2025 are available at https://nadi.dlnlp.ai/2025.

[2] Formerly the Workshop on Arabic Natural Language Processing, WANLP

Proceedings of The Third Arabic Natural Language Processing Conference, pages 720–733
November 8-9, 2025 ©2025 Association for Computational Linguistics

Arabic dialects remain under-studied due to limited resources—motivate a shift in NADI 2025 toward speech and diacritization.

NADI 2025 marks the *sixth* edition of the NADI shared task series, hosted by the Third Arabic Natural Language Processing Conference (ArabicNLP 2025[3]). In the following, we introduce several key new features that set it apart from previous versions, focusing on the challenges of real-world, spoken Arabic dialects:

A unified speech processing benchmark. This edition brings together three distinct but complementary tasks, "*dialect identification*", "*automatic speech recognition*", and "*diacritic restoration*", under one umbrella. This creates a comprehensive benchmark for evaluating system performance across the full spectrum of challenges in Arabic speech processing.

New evaluation datasets and unified benchmarking framework. We introduce a comprehensive suite of newly-curated datasets across all three subtasks. This includes a high-quality blind test sets *eight-hours* speech corpus for spoken dialect identification, a large-scale $10,807$ *utterances* for ASR, and a $1,332$ utterances for diacritic restoration, all covering diverse Arabic varieties. Beyond the data itself, NADI 2025 establishes a robust and unified evaluation framework featuring large-scale blind test sets to ensure fair comparison. This framework introduces novel paradigms, such as benchmarking model *adaptation* in the ADI task and offering distinct *open* and *closed* tracks for Diacritic Restoration.

A novel diacritic restoration task. We introduce the first shared task for diacritic restoration that moves beyond formal written Arabic (CA and MSA) to target *spoken dialects* and *code-switched language*. The task is uniquely designed to encourage multimodal solutions that leverage both speech and text as input.

Figure 1 provides a schematic overview of the NADI 2025 shared task, illustrating its three main subtasks including Spoken Arabic Dialect Identification, which covers *eight* regional dialects as "*Algerian*" (ALG), "*Egyptian*" (EGY), "*Emirati*" (UAE), "*Jordanian*" (JOR), "*Mauritanian*" (MAU), "*Moroccan*" (MOR), "*Palestinian*" (PAL), and "*Yemeni*" (YEM); Multidialectal Arabic ASR, which targets the exact same set of dialects; and Diacritic Restoration for Spoken Arabic Dialects, which encompasses MSA, mixed dialects, code-switched varieties, and CA.

The rest of the paper is organized as follows: Section 2 provides a review of related work on spoken Arabic processing and the history of the NADI shared task. In Section 3, we describe the NADI 2025 shared task in detail, including the three subtasks, their datasets, and evaluation metrics. Section 4 presents the results for all participating teams and baselines, followed by an overview of the submitted systems in Section 5. We conclude the paper in Section 6.

2 Literature Review

Unlike previous NADI tasks that relied on text, NADI 2025 concentrates on spoken Arabic dialects. Accordingly, this section covers related work on the subtasks of spoken language identification, ASR, and diacritic restoration. Before delving into the related work, it is useful to explore the history of NADI and its growth since its inception.

2.1 NADI Shared Task: Origins and Growth

`NADI-2020`, the first NADI shared task (Abdul-Mageed et al., 2020) involved two subtasks, one targeting country level (21 countries) and another focusing on province level (100 provinces), both exploiting X, *formerly Twitter*, data. NADI 2020 was the first shared task to exploit naturally occurring fine-grained dialectal text at the sub-country level.

`NADI-2021`, the second version (Abdul-Mageed et al., 2021) targeted the same 21 Arab countries and 100 corresponding provinces as NADI 2020, also using X data. However, it improved upon the previous version by removing non-Arabic data and distinguishing between MSA and dialectical Arabic (DA). It involved four subtasks: MSA-country, DA-country, MSA-province, and DA-province.

`NADI-2022` (Abdul-Mageed et al., 2022) continued the focus on studying Arabic dialects at the country level, but also included dialectal sentiment analysis with an objective to explore variation in socio-geographical regions that had not been extensively studied before.

`NADI-2023`, the fourth edition (Abdul-Mageed et al., 2023), proposed new machine translation subtasks from four dialectal Arabic varieties to MSA, in two themes (open-track and closed-track) as well as a dialect identification subtask at the country level.

[3] https://arabicnlp2025.sigarab.org/

Finally, NADI-2024, the fifth edition (Abdul-Mageed et al., 2024), targeted both dialect ID cast as a multi-label task, identification of the Arabic level of dialectness, and dialect-to-MSA machine translation.

2.2 Spoken Dialect Identification

Although CA and MSA have been extensively examined (Harrell, 1962; Badawi, 1973; Brustad, 2000; Holes, 2004), dialectal Arabic (DA) became the center of attention only relatively recently. A significant challenge in studying DA has been the scarcity of resources, prompting researchers to create new DA datasets targeting limited regions (Gadalla et al., 1997; Diab et al., 2010; Al-Sabbagh and Girju, 2012; Sadat et al., 2014; Harrat et al., 2014; Jarrar et al., 2016; Khalifa et al., 2016; Al-Twairesh et al., 2018; Alsarsour et al., 2018; Abu Kwaik et al., 2018; El-Haj, 2020; Haff et al., 2022; Nayouf et al., 2023; Jarrar et al., 2023). Several works introducing multi-dialectal datasets and models for region-level dialect identification (Zaidan and Callison-Burch, 2011; Elfardy et al., 2014; Bouamor et al., 2014; Meftouh et al., 2015) and the VarDial workshop series employing transcriptions of speech broadcasts (Malmasi et al., 2016) also followed. Other work developed relatively small-sized commissioned data (Bouamor et al., 2018; Salameh et al., 2018; Obeid et al., 2019).

Subsequently, larger datasets that cover between 10 to 21 countries were introduced (Mubarak and Darwish, 2014; Abdul-Mageed et al., 2018; Zaghouani and Charfi, 2018; Abdul-Mageed et al., 2020; Abdelali et al., 2021; Issa et al., 2021; Baimukan et al., 2022; Althobaiti, 2022; Elleuch et al., 2025; Hamad et al., 2025). The majority of these datasets are compiled from social media posts, especially X (formerly Twitter). More recently, benchmarks such as ORCA (Elmadany et al., 2023b) and DOLPHIN (Nagoudi et al., 2023) boast dialectal coverage.

Spoken dialect ID shares with text-based dialect ID a scarcity of labeled data. Important efforts to counter this include the introduction of the multi-genre and multidialectal ADI-5 (Ali et al., 2017) and ADI-17 (Ali et al., 2019; Shon et al., 2020) datasets (covering coarse regional and fine-grain country-level dialects, respectively). Moving from text to speech as a modality, however, introduces additional complexities such as potential channel mismatch between train and test sets due to differences in recording conditions, as is in the case with ADI-5 (Ali et al., 2017). Furthermore, dialect ID models may capture non-linguistic information such as gender and channel features (Chowdhury et al., 2020), and may experience major performance degradation in cross-domain and cross-dialect settings (Sullivan et al., 2023; Hamad et al., 2025).

2.3 Automatic Speech Recognition

Arabic ASR systems often struggle with dialectal speech, primarily due to lack of (or limited) dialectal data (Waheed et al., 2023). Mozilla Common Voice (Ardila et al., 2020) and MASC (MSA and Dialectal Speech) (Al-Fetyani et al., 2022) were introduced to alleviate this issue. However, both of these corpora label the data under a single label (Arabic) instead of different dialect names. Some of the audio and text samples in these datasets are also misaligned (Lau et al., 2025). The Casablanca project (Talafha et al., 2024) compiled high quality multidialectal speech for eight countries, providing a significant boost towards research in multidialectal ASR. Djanibekov et al. (2025) have also recently presented strong results for dialectal Arabic ASR as well as training strategies that work best based on data availability for each dialect.

2.4 Diacritic Restoration

Several text-based approaches (Alasmary et al., 2024; Elgamal et al., 2024; Fadel et al., 2019; Harrat et al., 2013) and resources (Toyin et al., 2025; Zerrouki and Balla, 2017) have been proposed for Arabic diacritic / vowel restoration. Aldarmaki and Ghannam (2023) found the speech based approach to outperform text only approaches. More recently, speech based or multi-modal approaches have also been proposed -albeit at a slow rate, mainly due to lack of parallel speech-text resources (Shatnawi et al., 2024). Elmadany et al. (2023a) report strong diacritization models as part of the Octopus toolkit, based on simple finetuning of AraT5 (Elmadany et al., 2022). Shatnawi et al. (2024) also propose an ASR-based diacritic restoration framework, where a pretrained ASR model generates vowelized transcripts refined by a secondary diacritization model. While their approach achieved high accuracy for CA, it fails to generalize to dialectal Arabic due to dataset limitations.

3 NADI 2025

NADI 2025 is the sixth edition of NADI shared task series. Since we extend the scope of the shared task to address broader challenges in multidialectal Arabic speech processing, we refer to NADI 2025 as "*the first multidialectal Arabic speech processing shared task*". This edition comprises three complementary subtasks: spoken Arabic dialect identification, multidialectal Arabic ASR, and diacritization restoration. Collectively, these subtasks target critical components of the Arabic speech technology pipeline, each addressing long-standing challenges arising from the language's rich dialectal variation, frequent code-switching, and the absence of diacritics in most written Arabic. By curating diverse, high quality datasets and establishing standardized evaluation protocols, NADI 2025 aims to catalyze the development of robust, generalizable systems that advance state of the art in Arabic speech and language processing.

3.1 Subtask 1 - Spoken Dialect Identification

Task Description. This subtask is an 8-way classification task to identify which of country-level dialect is being spoken in an utterance, with our set of countries being *Algeria, Egypt, Jordan, Mauritania, Morocco, Palestine, United Arab Emirates (UAE), and Yemen.*

Data. In this subtask, we follow similar procedure in selecting utterances to the data collection procedure of Casablanca (Talafha et al., 2024). For each dialect, different series were identified and the dialect spoken was verified by fluent speakers. For the Adaptation set, we utilize the same series as in Casablanca, but ensure there is no overlap with the series used for the Test set. By doing so, we aim to minimize the influence of potentially overlapping speakers, and to try to disentangle the dialect ID task from simple domain classification.

Evaluation Metric. We use both accuracy as well as the Language Recognition Evaluation 2022 average Cost metric (C_{avg}) (Lee et al., 2022). Because Cost is based on the probability of missed detections as well as false alarms for a given system it provides a complementary way to characterize model performance. At a high level, for two models that have similar accuracy but different Cost, the lower Cost model will providing a larger positive margin between the probability of the correct classes in comparison to incorrect classes, while the higher Cost model would have a smaller margin between correct and incorrect class probabilities.

3.2 Subtask 2 - Multidialectal Arabic ASR

Task Description. The ASR subtask2 in NADI-2025 focuses on building speech recognition systems that can handle spoken Arabic across a range of regional dialects: *Algerian, Egyptian, Jordanian, Mauritanian, Moroccan, Palestinian, Emirati, and Yemeni*. The task includes both monolingual and code-switched speech, which captures the variation speakers naturally use in different settings.

Data. The dataset used in this subtask is a subset of the Casablanca corpus (Talafha et al., 2024). In this subtask, we select balanced samples from each dialect. The training set is intended primarily for adaptation rather than full model training, encouraging participants to leverage transfer learning, domain adaptation, and other data-efficient strategies. We provide a total of $47,027$ utterances, evenly distributed across the eight dialects for the training, validation, and test sets ($1,600$ utterances per dialect per split). The only exceptions are Algeria, Palestine, and Yemen, which have 727, 900, and $1,180$ utterances, respectively, in the test set. These lower counts are due to the limited availability of samples for these dialects in the original Casablanca dataset.

Evaluation Metric. System performance is evaluated using the word error rate (WER) as the primary metric, reported both overall and per dialect. We also report character error rate (CER)[4] for additional insight into system performance, particularly for short utterances and morphologically rich forms. During evaluation, in line with Talafha et al., 2024, we apply a consistent text normalization pipeline to both system outputs and reference transcripts. Specifically, we: (a) retain only the % symbol, removing other special characters, (b) eliminate diacritics, (c) normalize Hamzas and Maddas to bare alif (ا), (d) convert Eastern Arabic numerals to Western Arabic numerals (e.g., ٢٩ becomes 29), and (e) preserve all Latin characters, as Casablanca contains code-switching segments in other languages.

Subtask 3 - Diacritic Restoration for Spoken Arabic Varieties. This subtask aims to advance

[4] In the case of a tie, we use the average CER as the tiebreaker.

Dataset	Type	Diacritized	Train	Dev	Test (Ours)
MDASPC	Multi-dialectal	True	60,677	—	5,164
TunSwitch	Dialectal, CS	True	5,212	165	110 (**110**)
ArzEn	Dialectal, CS	False	3,344	1,402	1,470 (**104**)
Mixat	Dialectal, CS	False	3,721	—	1,583 (**100**)
ClArTTS	CA	True	9,500	—	204
ArVoice	MSA	True	2,507	258	(**11**)
MGB2	MSA	False	—	—	5,365 (**40**)

Table 1: Number of sentences in datasets provided for the diacritic restoration sub-task. **Ours.** refers to the held-out test set for this shared task which we manually diaritize. **CA.** refers to Classical Arabic. **CS.** refers to code-switching.

research on automatic diacritic restoration for spoken Arabic varieties. As the vast majority of existing vowelization or diacritic restoration efforts focus on CA or MSA, we aim to raise attention to more challenging spoken varieties, such as dialects and code-switching, with a focus on generalization across different varieties. The objective of this subtask is to restore the diacritics of a given text. The text can be in a variety of forms, including MSA and Arabic dialects and may even include code-switched instances. In addition to text, all inputs have an associated speech utterance to encourage multi-modal approaches.

Data. This subtask encourages the development of multi-modal (speech + text) diacritic restoration models that generalize across Arabic variants. To enable the development of such models, we identified several high-quality data sets (Almeman et al., 2013; Abdallah et al., 2023; Al Ali and Aldarmaki, 2024; Hamed et al., 2020; Kulkarni et al., 2023; Toyin et al., 2025) of Arabic variants (CA, MSA, dialectal, CS) that include parallel speech and text. Table 1 shows a summary of the data sets provided to the participants for this subtask. The MDASPC dataset contains multi-dialectal speech with diacritized transcriptions and we include it for training. For the *TunSwitch* (Abdallah et al., 2023) training data, we used GPT-4o with a chain-of-thought prompt to initially diacritize the transcriptions. The diacritized output of GPT-4o was subsequently manually corrected with the corresponding audio as a reference by a native Arabic speaker. For code-switching, we provide undiacritized resources for training; *ArzEn* (Hamed et al., 2020), *Mixat* (Al Ali and Aldarmaki, 2024) and *MGB2* (Ali et al., 2016); for each dataset, we provide diacritized test sets by manually annotating random subsets of their test set.

Evaluation Metric. Similar to subtask 2, we use WER and CER as performance metrics for this subtask, which are chosen to enable the evaluation of diacritic restoration performance even for models that may change the underlying text, such as ASR-based or sequence-to-sequence models.

4 Shared Task Teams & Results

4.1 Participating Teams

A total of 44 teams registered for the NADI 2025. At the testing phase, a total of 100 valid entries were submitted by *eight* unique teams. The breakdown across the subtasks as follow: 34 submissions for subtask 1 by *five* teams, 47 submissions for subtask 2 by *six* teams and 19 submissions by *two* teams for subtask 3. Table 2 list NADI 2025 participated teams which completed the testing phase.

4.2 Baselines

We developed baseline (BL) models for each subtask to serve as reference points for evaluating the teams' systems. These models were not shared with participants during the competition.

Subtask 1. We finetune SpeechBrain's VoxLingua107 (Valk and Alumäe, 2021) ECAPA-TDNN (Desplanques et al., 2020) system[5] on the adaptation split of the dataset. We replace the classification layers of the pretrained system with new randomized layers corresponding to the smaller number of output classes (8); and train these new layers with the rest of the model frozen for 5K steps, and then unfreeze the model and train for an additional 25K steps. We use AdamW with base learning rate of $1e-4$, and apply a linear ramp up from $1/3$ base LR over 3K steps followed by constant LR until unfreezing, and then repeat the linear ramp up and plateau. Finally, Starting at 20K steps we applying an exponential decay.

Subtask 2. A zero-shot baseline is built on the pre-trained Whisper-Large-v3 model (Radford et al., 2022). Dialect-wise inference is performed on the official NADI 2025 subtask 2 ASR release available on Hugging Face[6], which provides validation splits for eight country-level dialects; official evaluation is conducted on a private Codabench test set. During inference, audio inputs are transcribed

[5] https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

[6] https://huggingface.co/datasets/UBC-NLP/NADI2025_subtask2_ASR

Team Name	Affiliation	Subtask
Abjad AI (Ghannam et al., 2025)	Abjad AI, Jordan & Saudi Arabia	1,3
BYZÖ (Abdullah et al., 2025)	Saarland University, Germany	2
Elyadata (Elleuch et al., 2025)	Elyadata, Tunisia	1,2
Hamsa	Hamsa	2
Lahjati (ALBawwab and Qawasmeh, 2025)	Princess Sumaya University, Jordan	1
MarsadLab (Attia et al., 2025)	Hamad Bin Khalifa University, Qatar	1,2
Munsit (Salhab et al., 2025b)	Lebanese American University, Lebanon	2
Unicorn (Elrefai, 2025)	Ain shams University, Egypt	3

Table 2: List of teams that participated in NADI 2025 shared task. Teams with accepted papers are cited.

using Whisper's default decoding parameters with language explicitly set to Arabic.

	Accuracy ↑	C_{avg} ↓
ELYDATA-LIA (Elleuch et al., 2025)	**79.8**	**17.88**
BYZÖ-ADI (Abdullah et al., 2025)	76.4	22.65
MarsadLab (Attia et al., 2025)	61.6	30.68
Abjad AI	61.2	34.77
Baseline	61.1	34.22
Lahjati (ALBawwab and Qawasmeh, 2025)	50.8	48.99

Table 3: Performance of the systems on the test set for Subtask 1. Results are sorted by Accuracy, while the average cost (C_{avg}) score is also reported, with lower values indicating better performance. The best performance is highlighted in bold.

Subtask 3. In this subtask, we provide three baselines: (I) A *text only* baseline based on the publicly available CATT model (Alasmary et al., 2024), which we use without further fine-tuning (II) an *ASR based* baseline where we use the ArTST v3 checkpoint (Djanibekov et al., 2025), which is pretrained on dialectal and code-switched Arabic, and finetune it for Arabic ASR with diacritics using the provided training data, and (III) a *multi-modal* diacritc restoration model designed as follows:

The raw waveform and corresponding transcriptions are passed in parallel to a speech and text encoder, respectively. The speech encoder is derived from ArTST ASR (Djanibekov et al., 2025), and the text encoder is derived from ArTST TTS model (Toyin et al., 2023). We then align the resulting text and speech embeddings using multi-head attention with 8 heads, followed by a trainable prediction component comprising 2 bi-directional LSTM layers, a 30% dropout layer, and a final linear prediction head to predict the corresponding diacritics. Simple ad-hoc post-processing is applied to add the predicted diacritics to the input text to produce the fully diacritized text output. This approach is inspired by the multi-modal diacritization model described in Shatnawi et al. (2024).

4.3 Results

Tables 3, 4, and 5, present the preformernce of the submitted systems on the test set for subtask 1, subtask 2, and subtask 3 respectively.

Subtask1. The ELYDATA-LIA team (Elleuch et al., 2025) achieved the best performance in terms of both accuracy and average cost C_{avg} (79.8 / 17.88), followed closely by BYZÖ-ADI (Abdullah et al., 2025) (76.4 / 22.65). Both top teams addressed the limited size of the Adaptation set in novel ways: ELYDATA-LIA leveraged the much larger ADI-20 dataset (Elleuch et al., 2025), while BYZÖ-ADI employed kNN voice conversion (Baas et al., 2023) to augment the training data with synthetic samples. In third place, MarsadLab (Attia et al., 2025) improved upon the baseline system through additional data augmentation and the introduction of an attention mechanism prior to the classification layer. In fourth place, Abjad AI fine-tuned a Whisper Small encoder with further data augmentation. While the third- and fourth-place systems were close in terms of accuracy (61.6 vs. 61.2), the approach by MarsadLa achieved a notably better C_{avg}, reducing it by approximately 4 points. Finally, we note that one team (Lahjati (ALBawwab and Qawasmeh, 2025)) perform below the baseline. Overall, these results highlight the effectiveness and diversity of data augmentation strategies.

Subtask 2. The Munist team (Salhab et al., 2025b) obtain the lowest overall average WER/CER scores (35.68/12.10) among all participating systems, achieving the best performance across all dialects except Moroccan, where it ranked second in both WER and CER, and Mauritanian, where it ranked first in CER and second in WER. The ELYADATA-LIA team (Elleuch et al., 2025) ranked second with scores of 38.52/14.52. They achieved the best performance on the

	Average	JOR	EGY	MOR	ALG	YEM	MAU	UAE	PAL
Munsit (Salhab et al., 2025b)	**35.68/12.20**	**20.68/5.64**	**20.88/7.33**	41.71/14.04	**53.62/18.44**	**44.62/14.30**	59.03/**23.28**	**22.66/6.55**	**22.27/8.05**
ELYADATA-LIA (Elleuch et al., 2025)	38.53/14.52	28.03/9.36	26.83/11.43	**38.26/13.66**	53.73/20.43	46.63/16.66	**58.10**/24.53	29.35/9.91	27.36/10.20
BYZÖ-Whisper (Abdullah et al., 2025)	39.78/14.75	28.84/9.47	29.50/11.91	43.06/15.52	55.04/20.59	46.41/16.05	59.36/24.84	28.38/9.04	27.65/10.59
Hamsa	42.04/16.18	32.24/9.90	24.72/10.21	48.21/18.11	60.32/23.33	51.76/20.41	66.23/29.11	28.00/8.98	24.87/9.41
BYZÖ-CTC (Abdullah et al., 2025)	44.14/15.58	31.74/9.94	37.23/12.57	43.31/15.07	56.12/21.38	46.14/15.68	63.32/26.70	38.65/11.14	36.62/12.18
Baseline	93.89/72.79	46.09/19.28	100.06/81.37	100.38/80.42	101.03/79.58	101.09/80.58	100.59/82.89	101.15/80.27	100.76/77.92
MarsadLab (Attia et al., 2025)	104.89/84.69	44.97/19.19	113.97/97.65	104.07/87.58	116.59/94.26	113.54/94.56	111.59/92.84	116.79/97.00	117.60/94.42

Table 4: Performance of the systems on the test set for Subtask 2. Results are sorted by the overall average WER/CER score across all dialects, with lower values indicating better performance. The best performance is highlighted in bold.

	WER ↓	CER ↓
Abjad AI (Ghannam et al., 2025)	**55**	**13**
Unicorn (Elrefai, 2025)	64	15
Baseline-I (*ASR based*)	88	45
Baseline-II (*text-only*)	65	16
Baseline-III (*multi-modal*)	66	16

Table 5: Performance of the systems on the test set for Subtask 3. Results are sorted by the overall average WER/CER score across all dialects, with lower values indicating better performance. The best performance is highlighted in bold.

Moroccan dialect (WER/CER of 38.26/13.66) and obtain the lowest CER for the Mauritanian dialect. Their performance on the Algerian dialect was only marginally lower than that of the first-ranked team, suggesting that their system demonstrates strong capabilities for North African dialects in general. The BYZÖ-Whisper team (Abdullah et al., 2025) ranked third, with average WER/CER scores of 39.78/14.75. The Hamsa team follow in fourth place, scoring 42.04/16.18, while the BYZÖ-CTC team (Abdullah et al., 2025) ranked fifth with 44.14/15.58. Only one team, MarsadLab (Attia et al., 2025), perform below the baseline, with notably higher average WER/CER scores of 104.89/84.69. The winning team Munist (Salhab et al., 2025b) surpassed the baseline by **58.21** WER points (93.89 → 35.68; ≈ 62% reduction). Furthermore, the variation in WER scores among the teams that surpassed the baseline is relatively low ($\sigma \approx 3.25$), corresponding to about 8.1% of the mean WER for these systems.

Subtask 3. The Abjad AI (Ghannam et al., 2025) perform the best with the lowest WER of 55% and CER of 13%. The Unicorn team (Elrefai, 2025) follow closely behind with WER of 64% and CER of 15%. Both teams improve over the provided baselines, the best of which achieve WER and CER of 65% and 16%, respectively.

5 Overview of Submitted Systems

In this section, we present an overview of the submitted systems for each subtask and summarize the methodological approaches adopted by the participating teams.

5.1 Subtask 1

ELYDATA-LIA (Elleuch et al., 2025). Using Whisper Large-v3 encoder as their base model, they adopt a two stage finetuning procedure to first finetune on the forthcoming ADI-20 dataset (Elleuch et al., 2025), and then use the NADI ADI Adaptation set for a second finetuning. Features of this approach include freezing the first 16 layers of the encoder and using plenty of data augmentation methods including speed perturbation, added noise, and frequency and chunk dropping.

BYZÖ-ADI (Abdullah et al., 2025) The authors choose a straightforward finetuning approach using w2v-BERT-2.0 (Barrault et al., 2023) model finetuned on the NADI ADI split (69% accuracy). However, in order to improve the robustness of the model, they add a data augmentation approach by using a voice conversion model (Baas et al., 2023) to re-synthesizing the training utterances using voice samples from the 4 Arabic speakers from the LibriVox project, and training on the mixed natural and synthetic audio, leading to their final model.

MarsadLab (Attia et al., 2025) Adopts a starting point similar to the baseline with a VoxLingua107 ECAPA-TDNN system that was finetuned on the ADI task. They introduce a number of features in the process including feature reweighting of the hidden representation just prior to the classification layer through the use of a lightweight attention mechanism, discriminative learning rate of the classification head, progressive unfreezing, as well as data augmentation using SpecAugment and injected noise.

Abjad AI Like the ELYDATA-LIA approach, this team used Whisper model, Whisper Small, and finetuned the encoder for dialect ID. They use only the NADI Adaptation set for finetuning, using SpecAugment (time and frequency masking) for data augmentation, and unfreezing the model partway through training.

Lahjati (ALBawwab and Qawasmeh, 2025) Using both the VoxLingua107 ECAPA-TDNN system as well as WavLM encoder, this fusion approach concattenates the outputs from the two models (WavLM pooled to match the ECAPA 256 dimension), and passes this combined representation through a layer normalization layer and then a two layers feedforward network to perform classification. Similar to other approaches the underlying models start frozen, with unfreezing at 8000 steps, followed by a ramp up, plateau, and then cosine annealing learning rate schedule.

5.2 Subtask 2

Munsit (Salhab et al., 2025b) This system follows a two-stage training pipeline combining large-scale weakly supervised pretraining and continual supervised fine-tuning, inspired by Salhab et al., 2025a. In the first stage, a Conformer-large model (Gulati et al., 2020) (121M parameters) was pretrained on 15K hours of weakly labeled Arabic speech, covering MSA and various dialects, with automatic labeling and no manual verification. In the second stage, the model was fine-tuned using a high-quality dataset composed of 3,000 hours of rigorously filtered weakly labeled data, excluding news content, and the official Casablanca Challenge training set, expanded via data augmentation. Training used the CTC (Graves et al., 2006) objective with a SentencePiece (Kudo and Richardson, 2018) vocabulary of 128 tokens, AdamW optimizer, Noam learning rate schedule, and dropout of 0.1, in a distributed setup across 8 NVIDIA A100 GPUs with bfloat16 precision. This approach enabled robust performance across all dialects, achieving the lowest average WER and CER in the shared task.

ELYADATA & LIA (Elleuch et al., 2025) For the ASR subtask, this team fine-tuned the SeamlessM4T-v2 (Barrault et al., 2023) Large Egyptian model separately for each of the eight dialects in the Casablanca dataset, producing eight distinct models. Training was performed for 6 epochs on NVIDIA A100 GPUs with a label-smoothed NLL loss (smoothing 0.2), AdamW optimizer, and a learning rate schedule with 100 warm-up steps ramping from 1e-9 to 5e-5. A batch size of 2 was used for all runs. This per-dialect fine-tuning approach yielded second overall in the shared task.

BYZÖ (Abdullah et al., 2025) The team submitted two independent systems. The first, `BYZÖ-Whisper`, fine-tuned the Whisper-Large-v3 model (Radford et al., 2023) (1.54B parameters) for Arabic dialect ASR using only the NADI shared task data, without external datasets or data augmentation. Text labels were preprocessed by removing bracketed content and normalizing spacing. Training followed a two-stage process: (1) domain adaptation on combined data from all dialects for 9000 steps (learning rate 1e-5, batch size 32), and (2) dialect-specific adaptation for 2000 steps per dialect using CER as the metric. The second, `BYZÖ-CTC`, fine-tuned the w2v-BERT-2.0 model (Barrault et al., 2023) (580M parameters) using a mix of public Arabic ASR datasets, then further fine-tuned per dialect on the shared task data (learning rate 1e-5, batch size 16). A multi-dialectal 3-gram Kneser-Ney smoothed language model, trained on collected dialect-specific text data, was integrated to reduce WER. This encoder-only CTC-based system was noted for efficiency and competitive zero-shot performance compared to Whisper large.

MarsadLab (Attia et al., 2025) For the ASR subtask, this team adopted Whisper-Large model (Radford et al., 2023) in a zero-shot setting, without any fine-tuning, preprocessing, or post-processing. Leveraging Whisper's multilingual capabilities, the system directly transcribed Arabic speech from multiple dialects in the test set. While the ECAPA-TDNN architecture was central to their ADI submission, it was not applied to ASR.

Hamsa Submissions were received from the `Hamsa` team; however, a system description was not made available.

5.3 Subtask 3

Unicorn (Elrefai, 2025) This team addressed the diacritic restoration task by fine-tuning the GEMM3N[7] multimodal model on both audio and text inputs. They formed diacritic restoration as a structured generation task where the model receives an undiacritized sentence and its corresponding audio and generates a fully diacritized ver-

[7] `https://unsloth.ai/`

sion. They fine-tuned with LoRA adaptation to efficiently adapt the model with the provided data for the sub-task only. They applied *nlpaug* for speech augmentation to simulate more diverse audio inputs. They perform inference by prompting GEMM3N with the raw audio and the corresponding undiacritized text.

Abjad AI (Ghannam et al., 2025) This team presented CATT-Whisper, which is a multimodal approach that combines both textual and speech information. Their model represents the text modality using an encoder extracted from their pre-trained model named CATT (Alasmary et al., 2024). The speech component is handled by the encoder module of the OpenAI Whisper base model (Radford et al., 2022). Their approach uses two integration strategies. The former consists of fusing the speech tokens with the input at an early stage, where the 1500 frames of the audio segment are averaged on the basis of 10 consecutive frames, resulting in 150 speech tokens only. To ensure embedding compatibility, these averaged tokens are processed through a linear projection layer prior to merging them with the text tokens. Contextual encoding is guaranteed by the CATT encoder module. The latter strategy relies on cross-attention, where text and speech embeddings are fused. Then, finally, the cross-attention output is fed to the CATT classification head for token-level diacritic prediction. They randomly deactivate the speech input during training for robustness, which allows the model to perform well with or without speech.

6 Conclusion

The *sixth* NADI shared task extends the scope of the series beyond text-based processing to encompass speech and diacritization, introducing three new subtasks: spoken dialect identification, Arabic ASR, and diacritic restoration. By releasing high-quality resources and providing clear evaluation protocols, our goal is to foster progress in inclusive Arabic speech processing. This edition, we received 44 registrations, with *eight* teams submitting system outputs and *seven* system description papers accepted. Results across the three subtasks highlight substantial headroom for improvement: even strong pretrained models continue to face challenges with multidialectal variability, code-switching, and diacritic restoration. We hope that this edition not only advances the state of the art on each individual subtask but also inspires future research toward unified, dialect-aware speech technologies for Arabic.

Limitations & Ethical Considerations

Despite the contributions of this year's shared task, several limitations remain across the three subtasks:

Coverage of dialects: Not all Arabic dialects are represented in the test sets, which limits the generalizability of results across the full dialect continuum.

Country-level labeling: We acknowledge that the use of country-level labels may be problematic. The continuum of Arabic dialects is complex, and using country affiliation as a stand-in for well-defined linguistic boundaries is not without limitations. This choice was made to ensure a reasonable degree of diversity in dialect coverage, while avoiding assumptions about the generalizability of models trained on a subset of dialects to unseen but related varieties.

Code-switching: The datasets capture only a limited subset of code-switching phenomena, whereas real-world Arabic speech often involves more diverse language mixing.

Real-world conditions: Background noise, spontaneous disfluencies, and accented speech are underrepresented in the datasets, limiting ecological validity.

Evaluation metrics: Metrics such as WER and CER may be misleading in the ASR task, since a dialectal utterance can often have multiple valid references. As the data provides only one reference per utterance, evaluation scores may underestimate system performance by penalizing alternative but correct transcriptions.

Acknowledgments

Muhammad Abdul-Mageed acknowledges support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[8] and UBC ARC-Sockeye.

[8] https://alliancecan.ca

References

Ahmed Amine Ben Abdallah, Ata Kabboudi, Amir Kanoun, and Salah Zaiem. 2023. Leveraging data collection and unsupervised learning for code-switched tunisian arabic automatic speech recognition. *Preprint*, arXiv:2309.11327.

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, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. Nadi 2023: The fourth nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2310.16117*.

Muhammad Abdul-Mageed, Amr Keleg, AbdelRahim Elmadany, Chiyu Zhang, Injy Hamed, Walid Magdy, Houda Bouamor, and Nizar Habash. 2024. Nadi 2024: The fifth nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2407.04910*.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. NADI 2020: The first nuanced arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop, WANLP@COLING 2020, Barcelona, Spain (Online), December 12, 2020*, pages 97–110. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. Nadi 2022: The third nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2210.09582*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, and Lyle Ungar. 2020. Toward micro-dialect identification in diaglossic and code-switched environments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5855–5876, Online. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim A. 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, WANLP 2021, Kyiv, Ukraine (Virtual), April 9, 2021*, pages 244–259. Association for Computational Linguistics.

Badr Abdullah, Yusser Al-Ghussein, Zena Al-Khalili, Ömer Özyilmaz, Matias Valdenegro-Toro, Simon Ostermann, and Dietrich Klakow. 2025. Saarland-groningen at nadi 2025 shared task: Effective dialectal arabic speech processing under data constraints. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. 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)*.

Maryam Khalifa Al Ali and Hanan Aldarmaki. 2024. Mixat: A data set of bilingual emirati-English speech. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 222–226, Torino, Italia. ELRA and ICCL.

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith A. Abandah, Adham Alsharkawi, and Maha Dawas. 2022. MASC: Massive arabic speech corpus. In *Proceedings of the 2022 IEEE Spoken Language Technology Workshop (SLT)*, page 1002206.

Rania Al-Sabbagh and Roxana Girju. 2012. Yadac: Yet another dialectal arabic corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2882–2889, Istanbul, Turkey. European Language Resources Association (ELRA).

Nora Al-Twairesh, Rawan N. Al-Matham, Nora Madi, Nada Almugren, Al-Hanouf Al-Aljmi, Shahad Alshalan, Raghad Alshalan, Nafla Alrumayyan, Shams Al-Manea, Sumayah Bawazeer, Nourah Al-Mutlaq, Nada Almanea, Waad Bin Huwaymil, Dalal Alqusair, Reem Alotaibi, Suha Al-Senaydi, and Abeer Alfutamani. 2018. Suar: Towards building a corpus for the saudi dialect. In *Fourth International Conference on Arabic Computational Linguistics, ACLING 2018*, volume 142 of *Procedia Computer Science*, pages 72–82, Dubai, United Arab Emirates. Elsevier.

Faris Alasmary, Orjuwan Zaafarani, and Ahmad Ghannam. 2024. Catt: Character-based arabic tashkeel transformer. *Preprint*, arXiv:2407.03236.

Sanad ALBawwab and Omar Qawasmeh. 2025. Lahjati at nadi 2025 a ecapa-wavlm fusion with multi-stage optimization. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Hanan Aldarmaki and Ahmad Ghannam. 2023. Diacritic recognition performance in arabic asr. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2023, pages 361–365.

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 279–284.

Ahmed Ali, Suwon Shon, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, James Glass, Steve Renals, and Khalid Choukri. 2019. The mgb-5 challenge: Recognition and dialect identification of dialectal arabic speech. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1026–1033. IEEE.

Ahmed Ali, Stephan Vogel, and Steve Renals. 2017. Speech recognition challenge in the wild: Arabic mgb-3. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 316–322. IEEE.

Khalid Almeman, Mark Lee, and Ali Abdulrahman Almiman. 2013. Multi dialect arabic speech parallel corpora. In *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6.

Israa Alsarsour, Esraa Mohamed, Reem Suwaileh, and Tamer Elsayed. 2018. Dart: A large dataset of dialectal arabic tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Maha J. Althobaiti. 2022. Creation of annotated country-level dialectal arabic resources: An unsupervised approach. *Natural Language Engineering*, 28(5):607–648.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC)*, pages 4218–4222.

Kais Attia, Md. Rafiul Biswas, Shimaa Ibrahim, Mabrouka Bessghaier, Firoj Alam, and Wajdi Zaghouani. 2025. Marsadlab at nadi: Arabic dialect identification and speech recognition using ecapa-tdnn and whisper. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Matthew Baas, Benjamin van Niekerk, and Herman Kamper. 2023. Voice conversion with just nearest neighbors. In *Interspeech 2023*, pages 2053–2057.

As-Said Muhámmad Badawi. 1973. *Mustawayat al-arabiyya al-muasira fi Misr*. Dar al-maarif.

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 (LREC 2022)*, pages 4586–4596, Marseille, France. European Language Resources Association (ELRA).

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, and 1 others. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1240–1245, Reykjavik, Iceland. European Language Resources Association (ELRA).

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Kristen Brustad. 2000. *The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian, and Kuwaiti Dialects*. Georgetown University Press.

Shammur A Chowdhury, Ahmed Ali, Suwon Shon, and James R Glass. 2020. What does an end-to-end dialect identification model learn about non-dialectal information? In *INTERSPEECH*, pages 462–466.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A Panoramic survey of Natural Language Processing in the Arab Worlds. *Commun. ACM*, 64(4):72–81.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. In *Interspeech 2020*, pages 3830–3834.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. Colaba: Arabic dialect annotation and processing. In *Proceedings of the LREC Workshop on Semitic Language Processing*, pages 66–74.

Amirbek Djanibekov, Hawau Olamide Toyin, Raghad Alshalan, Abdullah Alitr, and Hanan Aldarmaki. 2025. Dialectal coverage and generalization in arabic speech recognition. *Preprint*, arXiv:2411.05872.

Mahmoud El-Haj. 2020. Habibi – a multi dialect multi national arabic song lyrics corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association (ELRA).

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2014. Aida: Identifying code switching in informal arabic text. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 94–101, Doha, Qatar. 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.

Haroun Elleuch, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2025. ADI-20: Arabic Dialect Identification dataset and models. In *Interspeech 2025*, pages 2775–2779.

Haroun Elleuch, Youssef Saidi, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2025. Elyadata lia at nadi 2025: Asr and adi subtasks. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

AbdelRahim Elmadany, Muhammad Abdul-Mageed, and 1 others. 2022. Arat5: Text-to-text transformers for arabic language generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers)*, pages 628–647.

Abdelrahim Elmadany, Muhammad Abdul-Mageed, and 1 others. 2023a. Octopus: A multitask model and toolkit for arabic natural language generation. In *Proceedings of ArabicNLP 2023*, pages 232–243.

AbdelRahim Elmadany, ElMoatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023b. Orca: A challenging benchmark for arabic language understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9559–9586, Toronto, Canada. Association for Computational Linguistics.

Mohamed Lotfy Elrefai. 2025. Unicorn at nadi 2025 subtask 3: Gemm3n-dr: Audio-text diacritic restoration via fine-tuned multimodal arabic llm. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. 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.

Hassan Gadalla, Hanaa Kilany, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, Amr Shalaby, Krisjanis Karins, Everett Rowson, Robert MacIntyre, Paul Kingsbury, David Graff, and Cynthia McLemore. 1997. Callhome egyptian arabic transcripts ldc97t19. Web Download.

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 *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and 1 others. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + Baladi: Towards a Levantine Corpus. In *Proceedings of the International Conference on Language Resources and Evaluation(LREC 2022)*, Marseille, France.

Nagham Hamad, Mohammed Khalilia, and Mustafa Jarrar. 2025. Konooz: Multi-domain Multi-dialect Corpus for Named Entity Recognition. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 0–0, Vienna, Austria. Association for Computational Linguistics.

Injy Hamed, Ngoc Thang Vu, and Slim Abdennadher. 2020. ArzEn: A speech corpus for code-switched Egyptian Arabic-English. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4237–4246, Marseille, France. European Language Resources Association.

S. Harrat, M. Abbas, K. Meftouh, and K. Smaili. 2013. Diacritics restoration for arabic dialect texts. In *Interspeech 2013*, pages 1429–1433.

Salima Harrat, Karima Meftouh, Mourad Abbas, and Kamel Smaïli. 2014. Building resources for algerian arabic dialects. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association*, pages 2123–2127, Singapore. ISCA.

Richard S. Harrell. 1962. *A Short Reference Grammar of Moroccan Arabic: With Audio CD*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.

Clive Holes. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.

Elsayed Issa, Mohammed AlShakhori, Reda AlBahrani, and Gus Hahn-Powell. 2021. Country-level arabic dialect identification using rnns with and without linguistic features. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 276–281, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2016. Curras: An annotated corpus for the palestinian arabic dialect. *Language Resources and Evaluation*, pages 1–31.

Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2023. Lisan: Yemeni, Irqi, Libyan, and Sudanese Arabic Dialect Copora with Morphological Annotations. In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7. IEEE.

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).

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Ajinkya Kulkarni, Atharva Kulkarni, Sara Abedalmon'em Mohammad Shatnawi, and Hanan Aldarmaki. 2023. Clartts: An open-source classical arabic text-to-speech corpus. In *2023 INTERSPEECH*, pages 5511–5515.

Mingfei Lau, Qian Chen, Yeming Fang, Tingting Xu, Tongzhou Chen, and Pavel Golik. 2025. Data quality issues in multilingual speech datasets: The need for sociolinguistic awareness and proactive language planning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7466–7492, Vienna, Austria. Association for Computational Linguistics.

Yooyoung Lee, Craig Greenberg, Lisa Mason, and Elliot Singer. 2022. Nist 2022 language recognition evaluation plan.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and Arabic dialect identification: A report on the third DSL shared task. In *Proceedings of the third workshop on NLP for similar languages, varieties and dialects (VarDial3)*, pages 1–14.

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, Shanghai, China.

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, Ahmed El-Shangiti, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. Dolphin: A challenging and diverse benchmark for arabic nlg. *arXiv preprint arXiv:2305.14989*.

Amal Nayouf, Mustafa Jarrar, Fadi zaraket, Tymaa Hammouda, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian Arabic Dialects with Morphological Annotations. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*, pages 12–23. ACL.

Ossama Obeid, Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2019. Adida: Automatic dialect identification for arabic. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 6–11, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Fatiha Sadat, Farzindar Kazemi, and Atefeh Farzindar. 2014. Automatic identification of arabic language varieties and dialects in social media. In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, pages 22–27, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

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.

Mahmoud Salhab, Marwan Elghitany, Shameed Sait, Syed Sibghat Ullah, Mohammad Abusheikh, and Hasan Abusheikh. 2025a. Advancing arabic speech recognition through large-scale weakly supervised learning. *arXiv preprint arXiv:2504.12254*.

Mahmoud Salhab, Shameed Sait, Mohammad Abusheikh, and Hasan Abusheikh. 2025b. Munsit at nadi 2025 shared task 2: Pushing the boundaries of multidialectal arabic asr with weakly supervised pretraining and continual supervised fine-tuning. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. 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.

Suwon Shon, Ahmed Ali, Younes Samih, Hamdy Mubarak, and James Glass. 2020. Adi17: A fine-grained arabic dialect identification dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8244–8248. IEEE.

Peter Sullivan, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. On the robustness of arabic speech dialect identification. In *Interspeech 2023*, pages 5326–5330.

Bashar Talafha, Karima Kadaoui, Samar Mohamed Magdy, Mariem Habiboullah, Chafei Mohamed Chafei, Ahmed Oumar El-Shangiti, Hiba Zayed, Rahaf Alhamouri, Rwaa Assi, Aisha Alraeesi, and 1 others. 2024. Casablanca: Data and models for multidialectal arabic speech recognition. *arXiv preprint arXiv:2410.04527*.

Hawau Toyin, Amirbek Djanibekov, Ajinkya Kulkarni, and Hanan Aldarmaki. 2023. Artst: Arabic text and speech transformer. In *Proceedings of ArabicNLP 2023*, pages 41–51.

Hawau Toyin, Rufael Marew, Humaid Alblooshi, Samar M. Magdy, and Hanan Aldarmaki. 2025. ArVoice: A Multi-Speaker Dataset for Arabic Speech Synthesis. In *Interspeech 2025*, pages 4808–4812.

Jörgen Valk and Tanel Alumäe. 2021. Voxlingua107: a dataset for spoken language recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 652–658. IEEE.

Abdul Waheed, Bashar Talafha, Peter Sullivan, Abdelrahim Elmadany, and Muhammad Abdul-Mageed. 2023. Voxarabica: A robust dialect-aware arabic speech recognition system. In *Proceedings of ArabicNLP 2023*, pages 441–449.

Wajdi Zaghouani and Anis Charfi. 2018. Arap-tweet: A large multi-dialect twitter corpus for gender, age and language variety identification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 37–41, Portland, Oregon, USA. Association for Computational Linguistics.

Taha Zerrouki and Amar Balla. 2017. Tashkeela: Novel corpus of arabic vocalized texts, data for autodiacritization systems. *Data in Brief*, 11.

Munsit at NADI 2025 Shared Task 2: Pushing the Boundaries of Multidialectal Arabic ASR with Weakly Supervised Pretraining and Continual Supervised Fine-tuning

Mahmoud Salhab
CNTXT AI
Abu Dhabi, UAE
mahmoud.salhab@cntxt.tech

Shameed Sait
CNTXT AI
Abu Dhabi, UAE
shameed.ali@cntxt.tech

Mohammad Abusheikh
CNTXT AI
Abu Dhabi, UAE
mas@cntxt.tech

Hasan Abusheikh
CNTXT AI
Abu Dhabi, UAE
has@cntxt.tech

Abstract

Automatic speech recognition (ASR) plays a vital role in enabling natural human–machine interaction across applications such as virtual assistants, industrial automation, customer support, and real-time transcription. However, developing accurate ASR systems for low-resource languages like Arabic remains a significant challenge due to limited labeled data and the linguistic complexity introduced by diverse dialects. In this work, we present a scalable training pipeline that combines weakly supervised learning with supervised fine-tuning to develop a robust Arabic ASR model. In the first stage, we pretrain the model on 15,000 hours of weakly labeled speech covering both Modern Standard Arabic (MSA) and various Dialectal Arabic (DA) variants. In the subsequent stage, we perform continual supervised fine-tuning using a mixture of filtered weakly labeled data and a small, high-quality annotated dataset. Our approach achieves state-of-the-art results, ranking first in the multi-dialectal Arabic ASR challenge. These findings highlight the effectiveness of weak supervision paired with fine-tuning in overcoming data scarcity and delivering high-quality ASR for low-resource, dialect-rich languages.

1 Introduction

Automatic speech recognition (ASR), or speech-to-text (STT), converts spoken language into text, enabling voice-based interaction with machines (Algihab et al., 2019; Kheddar et al., 2024). ASR is widely applied in healthcare, robotics, law enforcement, telecommunications, smart homes, and consumer electronics, among other domains (Vajpai and Bora, 2016). Arabic, the fourth most used language online and one of the UN's six official languages, remains underrepresented in ASR research despite serving millions across 22 countries (Alwajeeh et al., 2014). Arabic exists in three forms: Classical Arabic (CA), the language of historical and religious texts; Modern Standard Arabic (MSA), used in formal contexts; and Dialectal Arabic (DA), comprising diverse regional variants (Al-Ayyoub et al., 2018). While some datasets, such as MASC (Al-Fetyani et al., 2021) and SADA (Alharbi et al., 2024), have advanced Arabic ASR, they remain limited in size and linguistic diversity, hindering model generalization. Neural ASR systems require vast transcribed datasets (Lu et al., 2020; Wang et al., 2021), but manual transcription is costly and time-intensive (Gao et al., 2023). We address this by proposing a weakly supervised Arabic ASR system based on the Conformer architecture (Gulati et al., 2020), trained on large-scale weakly labeled MSA and DA speech. In the first stage, we pretrain the model on 15,000 hours of weakly labeled speech covering both Modern Standard Arabic (MSA) and various Dialectal Arabic (DA) variants. In the subsequent stage, we perform continual supervised fine-tuning using a mixture of filtered weakly labeled data and a small, high-quality annotated dataset provided as part of the challenge (Talafha et al., 2025). This approach eliminates the need for extensive manual transcription and attains state-of-the-art results on the challenge's standard benchmarks, demonstrating the potential of weak supervision for low-resource languages.

2 Background

Arabic Automatic Speech Recognition (ASR) remains challenging due to data scarcity, lexical variation, morphological complexity, and dialect diver-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 734–739
November 8-9, 2025 ©2025 Association for Computational Linguistics

sity across 22 Arab countries (Ali et al., 2014; Cardinal et al., 2014; Diehl et al., 2012). Traditional systems often used hybrid HMM-DNN pipelines (Cardinal et al., 2014; Bouchakour and Debyeche, 2018). Dialectal variation is a major bottleneck, as most systems focus on Modern Standard Arabic (MSA) and high-resource dialects, performing poorly on low-resource varieties (Djanibekov et al., 2025). To address this, Djanibekov et al. released open-source ASR models covering 17 countries, 11 dialects, and code-switched Arabic-English/French speech. Other efforts integrate dialect identification directly into ASR (Waheed et al., 2023) or build dialect-specific systems, e.g., for Egyptian (Mousa et al., 2013) and Algerian Arabic (Menacer et al., 2017).

End-to-end architectures have advanced Arabic ASR by eliminating the need for intermediate feature extraction (Radford et al., 2023a). Notable examples include large-scale weakly supervised systems such as Whisper (Radford et al., 2023b). Weak supervision has proven particularly effective; for instance, (Salhab et al., 2025) trained a Conformer model from scratch on 15,000 hours of weakly labeled MSA and dialectal speech, achieving state-of-the-art results without relying on manual transcription.

3 Methodology

Our approach consists of two main stages: weakly supervised pretraining followed by continual supervised fine-tuning. In the first stage, we train the model on a large-scale, diverse speech dataset with weak labels—labels that are not guaranteed to be accurate (i.e., not manually verified)—in line with the strategy proposed in (Salhab et al., 2025).

In the second stage, the pretrained model is further fine-tuned using a smaller, high-quality dataset constructed from two main sources: (1) the official training data released for the task (the Casablanca training set (Talafha et al., 2024)), which is expanded through various augmentation techniques, including random Gaussian noise injection, background noise addition, and silence insertion; and (2) a filtered subset derived from the initial 15,000 hours of weakly labeled training data, selected through a rigorous data cleaning and filtering process.

An overview of the complete pipeline is presented in Figure 1. The following subsections provide a detailed explanation of each stage of the proposed approach.

3.1 Weakly Supervised Learning

Traditional supervised ASR training uses high-quality, human-annotated pairs (x_i, y_i), where the input x_i is typically a mel-spectrogram and the output y_i consists of a sequence of tokens, each selected from a predefined vocabulary. These accurate labels are assumed to be independently drawn from a clean data distribution, enabling the model to learn a function that performs well on unseen test examples. On the other hand, weakly supervised learning depends on automatically generated or crowd-sourced labels $\widehat{y}_i$, which may contain errors or noise. These weak labels come from a noisier distribution and might not precisely reflect the true transcription. Nonetheless, models trained on such data aim to generalize effectively when evaluated on clean datasets.

Building upon the approach introduced in (Salhab et al., 2025), we adopted the same training pipeline and experimental settings to develop the initial foundation model. Specifically, the model was trained on 15,000 hours of weakly annotated speech data, with automatic labeling performed using the same method described in the aforementioned work.

3.2 Continual supervised finetuning

In neural network-based ASR systems, training typically begins either from scratch—with randomly initialized weights and a large training corpus—or from a pretrained model that has already been exposed to a large-scale dataset. The latter approach enables faster convergence and often better generalization on the target task due to prior knowledge encoded in the pretrained weights.

In this stage, we adopt the second strategy by initializing the model with weights obtained from the first stage, which was trained on weakly labeled data. We then fine-tune this model using a smaller yet higher-quality dataset comprising 3,000 hours of filtered weakly annotated data. The filtering process was designed to exclude news content—largely composed of Modern Standard Arabic (MSA)—and to retain only segments that passed stringent quality thresholds, as outlined in the pipeline of (Salhab et al., 2025). Additionally, we incorporate the Casablanca Challenge training dataset, which is further expanded through various data augmentation techniques. Unlike the first stage that relied on noisy supervision, this fine-

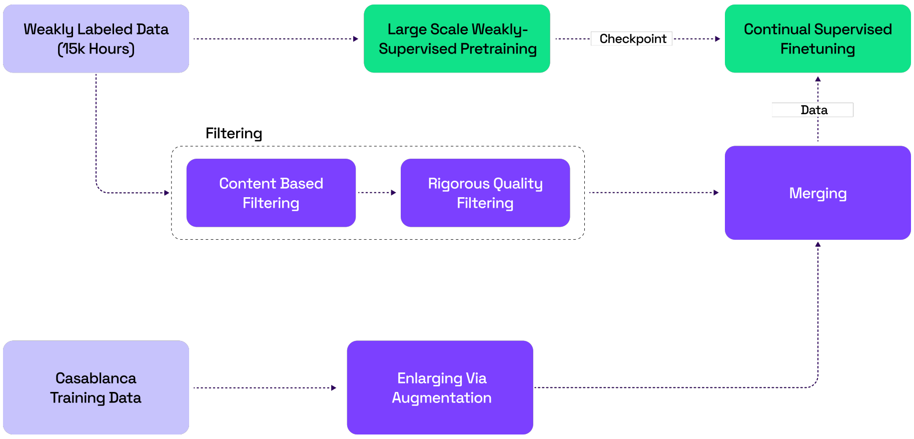

Weakly Labeled Data (15k Hours)
Large Scale Weakly-Supervised Pretraining
Checkpoint
Continual Supervised Finetuning
Data
Filtering
Content Based Filtering
Rigorous Quality Filtering
Merging
Casablanca Training Data
Enlarging Via Augmentation

Figure 1: The solution's full pipeline encompasses large-scale pretraining followed by continual fine-tuning.

tuning phase leverages only high-quality transcriptions.

3.3 Model Architecture

The Conformer architecture (Gulati et al., 2020) effectively models both long- and short-range dependencies in speech through a combination of convolutional modules and multi-head self-attention, making it highly suitable for automatic speech recognition. In this work, we adopt the same architecture as introduced in the original paper, specifically using the large variant of the model.

3.4 Experimental Setup

Our ASR experiments utilized the Conformer architecture trained with the Connectionist Temporal Classification (CTC) objective (Graves et al., 2006). To tokenize the transcripts, we employed a SentencePiece tokenizer (Kudo and Richardson, 2018) trained on the same training corpus, with a vocabulary of 128 tokens.

Model training was carried out in a distributed setting across 8 NVIDIA A100 GPUs using a global batch size of 512. Input features were 80-dimensional mel-spectrograms, extracted using a 25 ms frame length and a 10 ms hop size.

During the weakly supervised pretraining phase, optimization was performed using the AdamW optimizer combined with the Noam learning rate scheduler, incorporating 10,000 warm-up steps and peaking at a learning rate of 2×10^{-3}. For regularization purposes, we applied a dropout rate of 0.1 across all layers and used L2 weight decay. For the fine-tuning stage, the learning rate was reduced by a factor of ten.

To optimize training speed and reduce memory overhead, computations were performed using `bfloat16` precision. The Conformer model was initialized with random weights and comprised 18 encoder layers. Each layer featured a hidden dimension of 512, 8 attention heads, a convolutional kernel size of 31, and a feedforward expansion factor of 4. The complete model architecture contained approximately 121 million parameters.

3.5 Evaluation Metrics & Datasets

The model's performance was evaluated using Word Error Rate (WER) and Character Error Rate (CER). Training used a development set with paired speech and transcriptions, while testing involved blind evaluation on speech-only data via CodeBench.

4 Results

We evaluate our proposed system, against all participating teams using both Word Error Rate (WER) and Character Error Rate (CER) metrics, reported across multiple Arabic dialects. The results demonstrate the robustness of our approach across both evaluation and testing phases, as well as its ability to generalize across diverse dialectal variations.

As shown in Table 1, our system achieved the lowest average WER (35.69%), outperforming all other submissions. Notably, our work consistently maintained lower WER in most of the dialects, particularly excelling in Jordanian (20.68%), Egyptian (20.89%), and Emirati (22.67%) dialects. Similarly, Table 2 shows that our model achieved the lowest average CER (12.21%), with the best performance observed in Jordanian (5.64%) and Egyptian (7.33%) dialects. Tables 3 and 4 present a breakdown of WER and CER across evaluation and testing phases/datasets. The average WER decreased

Participant	Avg	JOR	EGY	MOR	ALG	YEM	MAU	UAE	PAL
msalhab96 (Ours)	**35.69**	20.68	20.89	41.72	53.62	44.62	59.03	22.67	22.28
youssef_saidi	38.54	28.03	26.83	38.27	53.73	46.63	58.11	29.35	27.36
yusser	39.78	28.84	29.50	43.07	55.04	46.42	59.37	28.38	27.66
alhassan10ehab	42.05	32.25	24.73	48.22	60.32	51.77	66.23	28.01	24.87
badr_alabsi	44.15	31.74	37.24	43.31	56.12	46.15	63.32	38.65	36.63
Baseline	93.90	46.10	100.07	100.38	101.03	101.09	100.59	101.15	100.77
rafiulbiswas	104.90	44.97	113.98	104.08	116.60	113.54	111.59	116.79	117.61

Table 1: Dialect-wise WER (%) Comparison Across Participants.

Participant	Avg	JOR	EGY	MOR	ALG	YEM	MAU	UAE	PAL
msalhab96 (Ours)	**12.21**	5.64	7.33	14.04	18.44	14.30	23.28	6.55	8.06
youssef_saidi	14.53	9.36	11.44	13.66	20.43	16.66	24.53	9.91	10.20
yusser	14.76	9.47	11.91	15.52	20.59	16.05	24.85	9.04	10.59
alhassan10ehab	16.19	9.90	10.21	18.12	23.34	20.41	29.11	8.99	9.41
badr_alabsi	15.59	9.95	12.57	15.07	21.39	15.69	26.70	11.15	12.19
Baseline	72.79	19.29	81.38	80.42	79.59	80.58	82.89	80.28	77.93
rafiulbiswas	84.69	19.19	97.66	87.59	94.27	94.56	92.85	97.01	94.42

Table 2: Dialect-wise CER (%) Comparison Across Participants.

Dialect	Evaluation	Testing
Avg	36.83	35.69
JOR	21.52	20.68
EGY	22.89	20.89
MOR	44.20	41.72
ALG	54.78	53.62
YEM	47.69	44.62
MAU	57.62	59.03
UAE	24.05	22.67
PAL	21.91	22.28

Table 3: Comparison of WER (%) Across Evaluation and Testing Datasets.

Dialect	Evaluation	Testing
Avg	11.94	12.21
JOR	5.39	5.64
EGY	7.50	7.33
MOR	14.06	14.04
ALG	17.71	18.44
YEM	14.73	14.30
MAU	21.73	23.28
UAE	6.97	6.55
PAL	7.40	8.06

Table 4: Comparison of CER (%) Across Evaluation and Testing Datasets.

from 36.83% during evaluation to 35.69% in testing, suggesting that our model generalizes well to unseen data. This trend is consistent across most dialects. For instance, the WER in the Jordanian dialect dropped from 21.52% to 20.68%, and in the Yemeni dialect from 47.69% to 44.62%.

Similarly, the average CER exhibited a slight increase from 11.94% (evaluation) to 12.21% (testing), though the variation across dialects remained minimal, underscoring the model's stability. These consistent results across both phases affirm the robustness and dialectal adaptability of our ASR system.

5 Conclusion

We present a scalable two-stage pipeline—pretraining on 15,000 hours of weakly labeled audio, then fine-tuning on a filtered 3,000-hour weak subset plus an augmented official training set—that, with data filtering, augmentation, and a Conformer backbone, achieved state-of-the-art performance and first place in the multi-dialectal Arabic ASR challenge, demonstrating that carefully curated weak supervision combined with targeted fine-tuning can overcome data scarcity and dialectal diversity.

References

Mahmoud Al-Ayyoub, Aya Nuseir, Kholoud Alsmearat, Yaser Jararweh, and Brij Gupta. 2018. Deep learning for arabic nlp: A survey. *Journal of Computational Science*, 26:522–531.

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith Abandah, Adham Alsharkawi, and Maha Dawas. 2021. Masc: Massive arabic speech corpus.

Wajdan Algihab, Noura Alawwad, Anfal Aldawish, and Sarah AlHumoud. 2019. Arabic speech recognition with deep learning: A review. In *Social Computing and Social Media. Design, Human Behavior and Analytics*, pages 15–31, Cham. Springer International Publishing.

Sadeen Alharbi, Areeb Alowisheq, Zoltán Tüske, Kareem Darwish, Abdullah Alrajeh, Abdulmajeed Alrowithi, Aljawharah Bin Tamran, Asma Ibrahim, Raghad Aloraini, Raneem Alnajim, Ranya Alkahtani, Renad Almuasaad, Sara Alrasheed, Shaykhah Alsubaie, and Yaser Alonaizan. 2024. Sada: Saudi audio dataset for arabic. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10286–10290.

Ahmed Ali, Hamdy Mubarak, and Stephan Vogel. 2014. Advances in dialectal arabic speech recognition: A study using twitter to improve egyptian asr. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Papers*, pages 156–162.

Ahmed Alwajeeh, Mahmoud Al-Ayyoub, and Ismail Hmeidi. 2014. On authorship authentication of arabic articles. In *2014 5th International Conference on Information and Communication Systems (ICICS)*, pages 1–6.

L Bouchakour and M Debyeche. 2018. Improving continuous arabic speech recognition over mobile networks dsr and nsr using mfccs features transformed. *International Journal of Circuits, Systems and Signal Processing*, 12:1–8.

Paul Cardinal, Ahmed Ali, Najim Dehak, Yifan Zhang, Takahiro A. Hanai, Yu Zhang, James R. Glass, and Stephan Vogel. 2014. Recent advances in asr applied to an arabic transcription system for al-jazeera. In *Proceedings of Interspeech 2014*, pages 2088–2092.

Frank Diehl, Mark JF Gales, Marcus Tomalin, and Philip C Woodland. 2012. Morphological decomposition in arabic asr systems. *Computer Speech & Language*, 26(4):229–243.

Amirbek Djanibekov, Hawau Olamide Toyin, Raghad Alshalan, Abdullah Alatir, and Hanan Aldarmaki. 2025. Dialectal coverage and generalization in Arabic speech recognition. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29490–29502, Vienna, Austria. Association for Computational Linguistics.

Dongji Gao, Hainan Xu, Desh Raj, Leibny Paola Garcia Perera, Daniel Povey, and Sanjeev Khudanpur. 2023. Learning from flawed data: Weakly supervised automatic speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. *Preprint*, arXiv:2005.08100.

Hamza Kheddar, Mustapha Hemis, and Yassine Himeur. 2024. Automatic speech recognition using advanced deep learning approaches: A survey. *Information Fusion*, page 102422.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *Preprint*, arXiv:1808.06226.

Liang Lu, Changliang Liu, Jinyu Li, and Yifan Gong. 2020. Exploring transformers for large-scale speech recognition. *arXiv preprint arXiv:2005.09684*.

Mohamed Amine Menacer, Odile Mella, Dominique Fohr, Denis Jouvet, David Langlois, and Kamel Smaïli. 2017. Development of the arabic loria automatic speech recognition system (alasr) and its evaluation for algerian dialect. *Procedia Computer Science*, 117:81–88. Arabic Computational Linguistics.

Amr Mousa, Hong-Kwang Kuo, Lidia Mangu, and Hagen Soltau. 2013. Morpheme-based feature-rich language models using deep neural networks for lvcsr of egyptian arabic. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8435–8439.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023a. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023b. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Mahmoud Salhab, Marwan Elghitany, Shameed Sait, Syed Sibghat Ullah, Mohammad Abusheikh, and Hasan Abusheikh. 2025. Advancing arabic speech recognition through large-scale weakly supervised learning. *Preprint*, arXiv:2504.12254.

Bashar Talafha, Karima Kadaoui, Samar Mohamed Magdy, Mariem Habiboullah, Chafei Mohamed Chafei, Ahmed Oumar El-Shangiti, Hiba Zayed, Mohamedou cheikh tourad, Rahaf Alhamouri, Rwaa

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, and 8 others. 2024. Casablanca: Data and models for multidialectal arabic speech recognition. *Preprint*, arXiv:2410.04527.

Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarar, Nizar Habash, and Muhammad Abdul-Mageed. 2025. Nadi 2025: The first multidialectal arabic speech processing shared task. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Jayashri Vajpai and Avnish Bora. 2016. Industrial applications of automatic speech recognition systems. *International Journal of Engineering Research and Applications*, 6(3):88–95.

Abdul Waheed, Bashar Talafha, Peter Sullivan, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. Voxarabica: A robust dialect-aware arabic speech recognition system. *arXiv preprint arXiv:2310.11069*.

Yongqiang Wang, Yangyang Shi, Frank Zhang, Chunyang Wu, Julian Chan, Ching-Feng Yeh, and Alex Xiao. 2021. Transformer in action: A comparative study of transformer-based acoustic models for large scale speech recognition applications. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6778–6782.

Lahjati at NADI 2025: A ECAPA-WavLM Fusion with Multi-Stage Optimization

Sanad Albawwab[1] **Omar Qawasmeh**[2]

[1]Knowledge Technologies Department, Applied AI Division,
Royal Scientific Society, Amman, Jordan
[2]Data Science Department, Princess Sumaya University for Technology, Amman, Jordan
Correspondence: sanad.bawwab@rss.jo, o.alqawasmeh@psut.edu.jo

Abstract

This paper presents Lahjati (ECAPA-WavLM Fusion with Multi-Stage Optimization) system for the spoken Arabic Dialect Identification (ADI) subtask at Nadi 2025 (Talafha et al., 2025), The task aims to automatically identify the dialect of spoken Arabic utterances, a challenging problem due to the rich linguistic diversity of Arabic and the scarcity of labeled speech resources. Our approach combines ECAPA-TDNN embeddings from SpeechBrain with WavLM-base representations. The proposed system achieved 94.08% accuracy on the validation set and ~51.0% on the test set. Challenges included differentiating acoustically similar dialect pairs and mitigating the effects of varied recording conditions, which likely contributed to performance degradation on unseen data. These findings highlight both the potential and limitations of fusing complementary speech representations for robust dialect identification.

1 Introduction

Arabic dialect identification from speech presents a significant challenge due to the extensive linguistic diversity across the Arab world and the scarcity of large, high-quality labeled datasets. These factors considerably hinder the development and optimization of speech technology applications. The task addressed here involves recognizing speech from eight distinct dialectal varieties: Palestinian, Yemeni, Mauritanian, Algerian, Moroccan, Jordanian, Egyptian, and Emirati Arabic. Achieving accurate identification of these dialects is essential for enhancing a wide range of downstream speech-processing technologies, including automatic speech recognition, machine translation, and conversational systems that can adapt effectively to regional linguistic variations.

This work builds upon the foundation laid by the Nuanced Arabic Dialect Identification (NADI) shared task, first introduced in 2020 (Abdul-Mageed et al., 2020), which provided standardized benchmark datasets and evaluation protocols for country-level Arabic dialect classification . The NADI initiative not only unified fragmented research efforts in this domain but also established a baseline for systematic comparison of approaches, paving the way for more fine-grained and context-aware dialect identification systems (Abdul-Mageed et al., 2021; Abdul-Mageed et al., 2022, 2023, 2024). By leveraging such frameworks and addressing current data limitations, this task aims to push the boundaries of robust, real-world Arabic speech dialect identification.

Main System Strategy

Our proposed system, **ECAPA_WavLM_Fusion**, integrates two complementary pretrained speech encoders to jointly capture *speaker-level* and *contextual acoustic* features, enabling robust Arabic dialect identification across eight dialect classes.

- **Speaker-Level Encoder:** The first component is **ECAPA-TDNN**, initialized from SpeechBrain's VoxLingua107 model, which generates 256-dimensional speaker embeddings directly from raw waveforms. This encoder excels at modeling speaker-specific timbre and prosodic characteristics, which are often correlated with dialectal traits.

- **Contextual Acoustic Encoder:** The second component is **WavLM-base** from Microsoft, a transformer-based model that produces 768-dimensional frame-level contextual embeddings. These embeddings are mean-pooled over time to obtain utterance-level representations, then linearly projected to 256 dimensions to match the ECAPA-TDNN feature space.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 740–744
November 8-9, 2025 ©2025 Association for Computational Linguistics

The outputs from both encoders are concatenated to form a 512-dimensional fused representation, which is `LayerNorm`-normalized and passed through a two-layer feedforward classifier with dropout regularization for final dialect prediction.

Training follows a two-stage fine-tuning strategy:

1. **Stabilization phase:** Encoder weights are frozen for the first 8,000 steps, allowing the classifier layers to learn stable decision boundaries from fixed embeddings.

2. **Full fine-tuning phase:** All parameters are unfrozen, and training continues with a multiphase learning rate scheduler consisting of linear warmup, constant hold, and cosine annealing.

We optimize with **AdamW**, incorporating weight decay for regularization and gradient clipping to mitigate exploding gradients.

By combining *speaker-discriminative* and *context-aware* representations, the **ECAPA_WavLM_Fusion** architecture effectively captures subtle phonetic and prosodic cues that differentiate closely related Arabic dialects, mitigating challenges posed by intra-dialect similarities. Code and pretrained models are available at our GitHub repository [1].

2 Background

This task addresses **Arabic dialect identification** directly from raw speech waveforms. The input consists of 16 kHz audio clips containing spoken Arabic from **eight dialects**: Palestinian, Yemeni, Mauritanian, Algerian, Moroccan, Jordanian, Egyptian, and Emirati. The system outputs a predicted dialect label corresponding to one of these classes. For example, given a short audio excerpt from a television program, the model must determine whether the speech is Egyptian, Moroccan, or one of the other target dialects.

The dataset used in this work comprises **high-quality multidialectal Arabic speech recordings** sampled at 16 kHz. It contains approximately **12,900 training samples** (∼8 hours of speech) and **12,700 validation samples** (∼8 hours), totaling around **16 hours of labeled audio**. An additional **8-hour blind test set** is provided for final evaluation.

A qualitative examination of the audio reveals that many clips are drawn from diverse media sources such as television dramas, movies, and talk shows, similar to the Casablanca dataset.(Talafha et al., 2024). This diversity introduces *natural conversational speech* with a wide range of acoustic conditions—including variations in background noise, recording quality, and speaker expressiveness—thereby creating a realistic and challenging benchmark for dialect classification.

Each audio sample is annotated with one of the eight target dialect labels, covering a spectrum of speech genres and speaker demographics. This diversity helps improve the robustness and generalization capabilities of trained models, making them more applicable to real-world settings.

Our system was developed for the **Spoken Arabic Dialect Identification (ADI)** track of the **Nuanced Arabic Dialect Identification (NADI)** shared task, which offers standardized datasets, clear evaluation protocols, and a competitive benchmarking platform for advancing research in fine-grained Arabic dialect recognition.

3 System Overview

Our system, `Lahjati` (`ECAPA_WavLM_Fusion`), leverages two complementary pretrained speech encoders to jointly capture *speaker-discriminative* and *context-aware* acoustic representations for Arabic dialect identification. This design aims to exploit both timbre/prosody cues (often linked to speaker identity and dialect) and broader contextual speech patterns for robust classification.

Key Architecture and Algorithms: The architecture integrates:

- **ECAPA-TDNN encoder** (Desplanques et al., 2020), initialized from the SpeechBrain VoxLingua107 model, which extracts 256-dimensional speaker embeddings from raw audio waveforms.

- **WavLM-base encoder** (Chen et al., 2022), a transformer-based model producing 768-dimensional contextual embeddings from frame-level speech representations, subsequently mean-pooled to form utterance-level features.

Both embeddings are linearly projected to 256 dimensions, concatenated into a unified 512-dimensional vector, normalized with `LayerNorm`, and passed through a two-layer feedforward neural

[1] `https://github.com/sanadbawab0/nadi2025/`

network with dropout regularization to predict one of eight target dialect classes.

The core forward computation of Lahjati can be formulated as:

$$\begin{aligned} \mathbf{e}_{\text{ecapa}} &= \text{ECAPA-TDNN}(x), \\ \mathbf{e}_{\text{wavlm}} &= \text{meanpool}(\text{WavLM}(x)), \\ \mathbf{h}_{\text{ecapa}} &= W_{\text{ecapa}}\,\mathbf{e}_{\text{ecapa}} + b_{\text{ecapa}}, \\ \mathbf{h}_{\text{wavlm}} &= W_{\text{wavlm}}\,\mathbf{e}_{\text{wavlm}} + b_{\text{wavlm}}, \\ \mathbf{h} &= \text{LayerNorm}\big([\mathbf{h}_{\text{ecapa}}; \mathbf{h}_{\text{wavlm}}]\big)\ , \\ \hat{y} &= \text{Classifier}(\mathbf{h}), \end{aligned} \tag{1}$$

where x is the input audio waveform and $\hat{y}$ is the predicted dialect label. The intermediate representations are defined as follows: $\mathbf{e}_{\text{ecapa}} \in \mathbb{R}^{256}$ is the ECAPA-TDNN speaker embedding, $\mathbf{e}_{\text{wavlm}} \in \mathbb{R}^{768}$ is the pooled WavLM contextual embedding, $\mathbf{h}_{\text{ecapa}}, \mathbf{h}_{\text{wavlm}} \in \mathbb{R}^{256}$ are the respective projected embeddings, and $\mathbf{h} \in \mathbb{R}^{512}$ is the fused representation after concatenation and normalization.

We employed pretrained ECAPA-TDNN and WavLM-base encoders, both initially frozen to exploit their rich acoustic representations while mitigating the risk of overfitting on the limited dialectal dataset. All experiments were conducted on the NADI 2025 dataset UBC-NLP/NADI2025_subtask1_SLID (UBC-NLP, 2025), available via Hugging Face.

Staged Fine-tuning Strategy: To address data scarcity and substantial dialectal overlap, we adopted a two-phase training procedure: (i) for the first 8,000 steps, encoder weights were frozen to allow the classifier to adapt to fixed embeddings; (ii) all parameters were then unfrozen, enabling joint fine-tuning with a multi-phase learning rate schedule (linear warmup, constant hold, cosine annealing). This approach balances early training stability with later model adaptability.

Training Pipeline: The training process follows five sequential steps: (1) raw audio is processed in parallel by ECAPA-TDNN and WavLM encoders; (2) embeddings are projected to a common 256-dimensional space, concatenated, and normalized; (3) a two-layer feedforward classifier produces logits for the eight target dialect classes; (4) cross-entropy loss is used for optimization; and (5) learning rate scheduling and gradient clipping are applied to ensure stable convergence.

Experimental Configurations: We compared training durations of 50,000 and 100,000 steps. Extending training to 100,000 steps improved validation accuracy by approximately $+2\%$, underscoring the benefits of prolonged fine-tuning for this task.

3.1 Experimental Setup

Data Splits We used the official splits released by the NADI 2025 organizers without modification. For Subtask 1 (SLID), we employed the UBC-NLP/NADI2025_subtask1_SLID dataset, comprising

- **Training:** 12,900 samples
- **Validation:** 12,700 samples

For the Subtask 1 (ADI) test phase, we used the UBC-NLP/NADI2025_subtask1_ADI_Test set containing 6,268 samples. No external data was incorporated.

Data Format Each instance consists of an audio recording and its label. In the SLID dataset, fields include id, audio, and country. In the ADI test set, fields include id and audio, where audio is stored as an array of float values along with the sampling rate.

Preprocessing Audio waveforms were loaded at their original sampling rate and resampled to 16 kHz using the AutoFeatureExtractor from the Hugging Face microsoft/wavlm-base model. Dialect labels were mapped to integer IDs via:

$$\text{labels2id} = \{\text{country} : \text{index}\}.$$

For batch preparation:

- Raw waveforms were padded to the longest sequence in the batch for ECAPA-TDNN input.
- WavLM inputs were prepared using AutoFeatureExtractor (return_tensors="pt", padding=True) with a 16 kHz sampling rate.

No data augmentation was applied.

Batching Training batches comprised 4 audio samples each, randomly shuffled for training and kept in sequential order for validation.

Training Hyperparameters Models were trained for up to 100,000 steps using AdamW with a weight decay of 10^{-2}. Learning rates were set to 1×10^{-5} for ECAPA-TDNN and WavLM encoder parameters, and 1×10^{-4} for the projection layers and classifier. Gradient clipping (max norm = 1.0) was applied to mitigate exploding gradients.

Freezing Strategy & Learning Rate Schedule Pretrained encoders were frozen for the first 8,000 steps, followed by full-network fine-tuning. The learning rate schedule, implemented via PyTorch, consisted of:

1. **Frozen phase (0–8,000 steps):**
   - *Warmup* (0–3,000): `LinearLR`, start factor $= \frac{1}{3}$.
   - *Constant* (3,000–8,000): `ConstantLR`.
2. **Unfrozen phase (8,000–100,000 steps):**
   - *Warmup* (8,000–12,000): `LinearLR`, start factor $= \frac{1}{10}$.
   - *Constant* (12,000–52,000): `ConstantLR`.
   - *Cosine decay* (52,000–100,000): `CosineAnnealingLR`.

Evaluation Metrics System performance was assessed using two metrics:

- **Accuracy:** Proportion of correctly classified samples over the total number of evaluated samples.
- **Average Cost:** Following the NIST LRE 2022 formulation (Lee et al., 2022), log-likelihood ratios were computed from model logits via pairwise class comparisons to estimate prediction confidence. The cost combines false positive rate (FPR) and false negative rate (FNR) as:

  $$\text{Cost} = \text{FPR} + \text{FNR}.$$

  This metric balances penalties for missed detections and false alarms across varying decision thresholds.

Accuracy served as the primary metric, with average cost providing a complementary error-sensitive measure.

4 Results

Experiments were conducted on the official NADI 2025 validation and blind test splits.

Validation: Our system achieved **94.08%** accuracy with an average NIST cost of **6.37%**, ranking 3rd on the validation leaderboard.

Test: Performance dropped to **~51.0%** accuracy with an average NIST cost of **~49.0%**, likely due to domain mismatch and data distribution shifts between validation and test sets.

No additional ablation or error analysis was performed; results focus on the primary leaderboard metrics.

5 Conclusion

We presented `Lahjati` (`ECAPA_WavLM_Fusion`), a dual-encoder fusion model combining ECAPA-TDNN and WavLM to jointly capture speaker-level and contextual acoustic representations for Arabic dialect identification. A staged training regime—initial encoder freezing followed by fine-tuning—yielded competitive results, with a validation accuracy of 94.08% on the NADI 2025 dataset.

Limitations include the absence of data augmentation and challenges from dialectal overlap, both of which may hinder generalization to unseen data. Future work will investigate advanced augmentation, alternative fusion architectures, and hyperparameter optimization to improve robustness.

This study offers a competitive, reproducible benchmark for Arabic dialect identification, contributing toward improved speech processing for underrepresented language varieties.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. Nadi 2023: The fourth nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2310.16117*.

Muhammad Abdul-Mageed, Amr Keleg, AbdelRahim Elmadany, Chiyu Zhang, Injy Hamed, Walid Magdy, Houda Bouamor, and Nizar Habash. 2024. Nadi 2024: The fifth nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2407.04910*.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. NADI 2020: The first nuanced arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop, WANLP@COLING 2020, Barcelona, Spain (Online), December 12, 2020*, pages 97–110. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. Nadi 2022: The third nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2210.09582*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim A. 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, WANLP 2021, Kyiv, Ukraine (Virtual),* *April 9, 2021*, pages 244–259. Association for Computational Linguistics.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*.

Yooyoung Lee, Craig Greenberg, Lisa Mason, and Elliot Singer. 2022. Nist 2022 language recognition evaluation plan.

Bashar Talafha, Karima Kadaoui, Samar Mohamed Magdy, Mariem Habiboullah, Chafei Mohamed Chafei, Ahmed Oumar El-Shangiti, Hiba Zayed, Rahaf Alhamouri, Rwaa Assi, Aisha Alraeesi, and 1 others. 2024. Casablanca: Data and models for multidialectal arabic speech recognition. *arXiv preprint arXiv:2410.04527*.

Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarar, Nizar Habash, and Muhammad Abdul-Mageed. 2025. Nadi 2025: The first multidialectal arabic speech processing shared task. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

UBC-NLP. 2025. Nadi 2025 subtask 1: Spoken language identification (slid) dataset. https://huggingface.co/datasets/UBC-NLP/NADI2025_subtask1_SLID. Accessed 2025-08-11.

Saarland-Groningen at NADI 2025 Shared Task: Effective Dialectal Arabic Speech Processing under Data Constraints

Badr M. Abdullah[1,5] **Yusser Al Ghussin**[1,2] **Zena Al-Khalili**[1,5] **Ömer Tarik Özyilmaz**[3,4]
Matias Valdenegro-Toro[4] **Simon Ostermann**[1,2] **Dietrich Klakow**[1,5]

[1]Saarland University, Germany [2]German Research Center for AI (DFKI), Germany
[3]University Medical Center Groningen, University of Groningen, The Netherlands
[4]University of Groningen, The Netherlands
[5]SFB 1102 - Information Density and Linguistic Encoding (IDeaL)

Abstract

We present our systems for the NADI 2025 shared task on multidialectal Arabic speech processing, participating in both spoken dialect identification (ADI) and automatic speech recognition (ASR) subtasks. Working under data constraints by using only the provided shared task resources for dialect adaptation, we explore effective model adaptation strategies for dialectal Arabic speech. For ADI, we fine-tune w2v-BERT 2.0 and employ voice conversion as data augmentation, improving accuracy from 68.71% to 76.40% on a blind cross-domain test set. For ASR, we develop two complementary approaches: (1) a CTC-based model pre-trained on public Arabic speech data, and (2) Whisper-based models using two-stage fine-tuning. Our experiments show that while dialect-centric CTC models exhibit better zero-shot dialectal performance (58.89 vs 93.90 WER), Whisper achieves better performance after dialect-specific adaptation, which reduces WER from 93.89 to 39.78 WER. We also demonstrate that using character error rate (CER) as a validation criterion provides practical benefits with minimal performance tradeoffs. Despite using no external resources for dialect adaptation beyond the shared task data, our systems ranked second in ADI and third in ASR, demonstrating that careful adaptation strategies can overcome data constraints in dialectal speech processing.

1 Introduction

The Arabic language exhibits a rich linguistic variation landscape. While Modern Standard Arabic (MSA) serves as the official language and codified variety across all Arabic-speaking countries, it primarily exists in formal situations such as scripted news broadcasts and official documents. Daily spoken communication occurs exclusively in regional dialects that differ from MSA and each other at every linguistic level: prosody, phonology, lexicon, and syntax. Although spoken dialects still lack a standardized orthography and are not formally taught in schools, they maintain a strong cultural presence through songs, folktales, and cinema (Holes, 2004; Habash, 2010).

Despite recent advances in language technology, MSA remains the only Arabic variety that is well-supported by AI-powered speech technology. For example, while state-of-the-art ASR systems (e.g., Radford et al. (2023)'s Whisper model) work well on MSA speech, they fail to adequately transcribe and translate dialectal speech. To address this gap, recent community efforts have focused on building speech resources for Arabic dialects. Notable among these is the Casablanca corpus (Talafha et al., 2024), the largest fully supervised Arabic speech dataset covering eight regional dialects. The NADI 2025 shared task builds on this resource to advance speech technologies for Arabic dialects across three speech processing subtasks.

In the NADI 2025 shared task, we participated in two subtasks: spoken Arabic dialect identification (ADI) and multidialectal Arabic ASR. Working exclusively with the provided datasets by the organizers, we explored which model adaptation techniques are most effective under resource constraints. For ADI, we adapted the multilingual pretrained w2v-BERT 2.0 model using supervised fine-tuning and voice conversion as audio augmentation. We found that this approach improves robustness to domain mismatch, which is consistent with our prior work (Abdullah et al., 2025). For ASR, we developed two systems: (1) a dialect-centric model based on connectionist temporal classification (CTC) loss and (2) fine-tuned Whisper models. While the dialect-centric approach performed better in zero-shot settings, dialect-specific Whisper-based models achieved superior performance after fine-tuning. Overall, our best ADI system ranked second while our best ASR system ranked third in their respective subtasks, despite our data constrained setup.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 745–751
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Shared Task Description

The NADI shared task series has evolved significantly over the years, with previous iterations (2020-2024) focusing primarily on text-based dialect identification at various granularities (Abdul-Mageed et al., 2020, 2021; Abdul-Mageed et al., 2022, 2023, 2024). NADI 2025 represents a major shift to speech processing, recognizing that dialectal variation is most naturally expressed in spoken form and that speech technology lags behind text processing for Arabic dialects.

The NADI 2025 shared task focuses on advancing multidialectal Arabic speech processing through three complementary subtasks that address critical challenges in dialect-aware speech technology (Talafha et al., 2025). Building on the Casablanca corpus (Talafha et al., 2024), the task provides participants with resources for eight Arabic dialects throughout the Middle East and North Africa. The dataset covers eight country-level dialects with the following abbreviations used throughout this paper: Algerian (ALG), Egyptian (EGY), Emirati (UAE), Jordanian (JOR), Mauritanian (MAU), Moroccan (MOR), Palestinian (PAL), and Yemeni (YEM).

2.1 Subtask 1: Spoken Arabic Dialect Identification (ADI)

This subtask requires systems to predict the spoken Arabic dialect from short audio clips. Given the rich linguistic diversity of Arabic and the limited availability of labeled dialectal speech data, accurate dialect identification remains challenging, especially in domain mismatch settings (Sullivan et al., 2023; Abdullah et al., 2025). This subtask aims to evaluate how well modern multilingual speech models and embedding techniques can distinguish between dialectal variations using acoustic-phonetic features. The provided dataset for this subtask consists of dialect-annotated speech samples for three splits: adaptation, validation, and test, where each split is 8 hours of speech.

2.2 Subtask 2: Multidialectal Arabic ASR

In this subtask, participants are required to develop ASR systems capable of adequately transcribing speech across multiple Arabic dialects. The primary challenge lies in handling the substantial phonological, lexical, and syntactic variations between dialects while maintaining high-quality transcriptions across all varieties. Systems are evaluated using both Word Error Rate (WER) and Character Error Rate (CER) metrics, which measure the extent to which ASR generated transcripts match gold human transcriptions. The provided dataset for this subtask consists of transcribed speech samples for three splits: adaptation (12,800 utterances), validation (12,800 utterances), and test (10,298 utterances).

3 System Overview

In this section, we describe our systems for the shared task. We refer to all our systems under the name **BYZÖ**, an acronym formed from the first letters of each core team member's first name.

3.1 Spoken Arabic Dialect Identification

We fine-tuned the multilingual pre-trained speech model w2v-BERT-2.0 for ADI using only the provided shared task data. We add an 8-way classification head that is randomly initialized on top of the pre-trained model for this task. To improve the model's robustness against unpredictable recording variations, we used k-nearest neighbor (k-NN) voice conversion (Baas et al., 2023) to create resynthesized samples from the training data using target voices from LibriVox audiobook recordings. We used four target voices from LibriVox who spoke standard Arabic. Using this approach, we created synthesized data that is four times larger than the original dataset. Our results show that using a combined dataset (natural + resynthesized) significantly improves performance without adding any natural samples or requiring architectural modifications.

3.2 Multidialectal Arabic ASR

3.2.1 System 1: `BYZÖ-whisper`

Similar to prior research in dialectal Arabic ASR using Whisper (Özyilmaz et al., 2025), we fine-tune the Whisper-large-v3 model for multidialectal Arabic ASR and examine how different training strategies affect its performance. Our Whisper-based approach consists of three aspects:

1. Two-stage fine-tuning procedure. First, we perform domain adaptation by fine-tuning all model layers on the combined dataset from all dialects, creating a domain-adapted multidialect baseline. Second, we conduct dialect adaptation by fine-tuning eight dialect-specific models, each trained exclusively on its respective dialect data using the same configuration. This approach combines the benefits of shared dialectal knowledge with dialect-specific optimization.

2. Alternative validation criterion. We experiment with CER as an alternative validation metric to stop early during dialect adaptation. While domain adaptation uses WER for validation, we compare WER versus CER as stopping criteria for dialect-specific fine-tuning. Using CER for early stopping may prevent overfitting to frequent word patterns and yield better character-level performance.

3. Parameter-efficient fine-tuning via LoRA. We also experiment with Low-Rank Adaptation, or LoRA (Liu et al., 2024), as an efficient alternative to full fine-tuning. LoRA inserts trainable rank-decomposition matrices into the model's weight layers while keeping original weights frozen, reducing computational costs and potential overfitting on limited dialect data.

3.2.2 System 2: `BYZÖ-ctc`

As an alternative to Whisper-based models, we developed our own dialect-centric ASR model by fine-tuning w2v-BERT-2.0 (580M parameters) with CTC loss. The model underwent two-stage training: (1) supervised fine-tuning on public Arabic ASR datasets including Arabic Common Voice (Ardila et al., 2020), SADA (Alharbi et al., 2024), Linto (Abdallah et al., 2024; Naouara et al., 2025), D-Voice 2.0 (Allak et al., 2021), and the Egyptian Arabic ASR dataset on Kaggle, and (2) dialect-specific fine-tuning using only the shared task data. This encoder-only architecture is more efficient than Whisper-based models and we show that it outperforms Whisper-large in zero-shot settings.

To enhance the dialectal fidelity of ASR output, we trained dialect-specific n-gram language models with Kneser-Ney smoothing (with $n = 3$) using curated text corpora for each dialect. These LMs were integrated into BYZÖ-ctc's decoding to constrain acoustically plausible but linguistically unlikely word sequences, reducing grammatical and lexical errors in the final transcriptions. The LMs training corpora are detailed in Appendix B.

4 Experimental Setup

We used the Hugging Face Transformers library and the Trainer module to fine-tune our ASR and ADI systems. For our Whisper-based systems, we used the AdamW optimizer with a linear learning rate warmup for 500 steps to a peak of 1×10^{-5}, followed by cosine decay. Each model was trained for up to 2000 steps. For our w2vBERT 2.0-based systems, we used the AdamW optimizer with a linear learning rate warmup for 10% of the adaptation samples to a peak of 1×10^{-5}, followed by linear decay. We applied minimal text processing to the text transcripts for the ASR systems. We share our code and models for reproducibility[1].

System	Accuracy (%)	Avg. Cost
Baseline	61.09	0.342
BYZÖ-ADI	68.71	1.136
BYZÖ-ADI + VC	**76.40**	**0.227**

Table 1: Dialect identification performance metrics. Our approach with voice conversion (VC) achieves optimal performance with 76.4% accuracy (higher is better) as well as the lowest cost value (lower is better).

5 Experimental Results

5.1 Spoken Arabic Dialect Identification

Table 1 presents the ADI results on the NADI 2025 test set with two evaluation metrics: accuracy and average cost as define by the NIST Language Recognition Evaluation campaign. The baseline system, which is based on a Pretrained ECAPA-TDNN VoxLingua107 system fine-tuned on adaptation split, achieves 61.09% accuracy with a cost of 0.342. Our initial BYZÖ-ADI model improves accuracy to 68.71%, though at a higher cost of 1.136, indicating increased confusion between dialects. However, incorporating voice conversion (VC) as a data augmentation strategy yields substantial improvements on both metrics. The BYZÖ-ADI + VC system achieves the best performance with 76.40% accuracy while simultaneously reducing the cost to 0.227. This 7.69 percentage point improvement in accuracy over the base model demonstrates that voice conversion effectively enhances the model's robustness to acoustic variations while improving its discriminative ability across dialects.

5.2 Multidialectal Arabic ASR

Table 2 shows WER results across eight dialects. The zero-shot Whisper baseline fails completely on dialectal speech with an average WER of 93.90, except for Jordanian (46.10). Our BYZÖ-ctc model performs outperforms Whisper in zero-shot set-

[1] https://github.com/Yusser95/NADI-NLP-2025-Whisper

System	ALG	EGY	JOR	MAU	MOR	PAL	UAE	YEM	AVG
Whisper (Zero-shot)	101.0	100.1	46.09	100.6	100.4	100.8	101.6	101.1	93.89
BYZÖ-ctc (Zero-shot)	75.17	48.40	40.67	81.25	72.21	52.24	46.91	54.23	58.89
BYZÖ-ctc + SFT	60.82	40.59	44.52	67.00	50.74	45.45	42.31	49.24	50.08
BYZÖ-ctc + SFT + LM	57.12	35.23	32.62	62.81	45.46	37.32	38.20	46.42	44.40
BYZÖ-whisper + SFT I	65.10	32.88	31.49	69.80	57.80	31.31	35.69	53.14	47.15
BYZÖ-whisper + SFT II	**55.04**	**29.50**	**28.84**	**59.37**	**43.07**	**27.66**	**28.38**	**46.42**	**39.78**

Table 2: Word Error Rate (WER) performance across eight Arabic dialects on the NADI 2025 test set. All our systems used only shared task data for dialect adaptation. The baseline Whisper zero-shot results demonstrate the challenge of dialectal ASR, while our BYZÖ systems show progressive improvements. Best results (in bold) are achieved by two-stage fine-tuning with CER as criterion. Lower values indicate better performance.

tings (WER of 58.89), showing that dialect-centric pre-training is effective for dialectal speech. After dialect-specific fine-tuning, both our systems improve significantly. The CTC model reduces average WER from 58.89 to 50.08 with supervised fine-tuning, and further to 44.40 when adding language models. The Whisper-based models achieve better final results despite worse zero-shot performance. Whisper fine-tuning gives a WER of 47.15, while two-stage fine-tuning with CER as a validation criterion achieves the best performance at 39.78. This 4.62 point gap suggests the encoder-decoder architecture handles dialectal variations better than CTC when properly fine-tuned. Both models show the largest gains on low-resource dialects like Mauritanian and Moroccan, reducing WER by over 40 points from baseline.

On the other hand, Table 3 shows the performance measured by CER for the eight dialects. Interestingly, the model that yields the lowest WER for a dialect does not necessarily yield the lowest CER. This finding suggests that WER and CER might capture different model competences and therefore should be combined when evaluating ASR models.

6 Discussion

Our results reveal several important insights about adapting ASR systems for dialectal Arabic speech. The dramatic failure of zero-shot Whisper (93.9 WER average) highlights a fundamental challenge: models trained primarily on MSA and high-resource languages cannot generalize to Arabic dialects, despite Whisper's multilingual capabilities. This performance gap shows how the distinct phonological and lexical features, which separate dialectal Arabic varieties from MSA, affect the performance of ASR systems. The success of our adaptation strategies raises interesting questions about model architecture choices. While our CTC-based model shows better zero-shot dialectal speech-to-text transcription (58.89 vs 93.90 WER), the Whisper architecture ultimately achieves superior performance after fine-tuning (39.78 WER). This suggests that encoder-decoder models may have greater capacity for dialectal adaptation when provided with adequate supervision for each dialect, possibly due to their ability to model longer-range dependencies and contextual information during decoding.

7 Conclusion

We presented data-constrained approaches for the NADI 2025 shared task, achieving competitive results in both dialect identification and ASR subtasks. Our key findings include: (1) voice conversion improves ADI accuracy by 7.69 percentage points while reducing classification uncertainty, (2) dialect-centric pre-training provides better zero-shot performance than general multilingual models, and (3) two-stage fine-tuning with character-level optimization yields the best ASR results. Our experiments reveal important architectural trade-offs. CTC models offer better initial dialectal understanding and efficiency, while encoder-decoder architectures show superior adaptation capacity after fine-tuning. Future work should address the persistent performance disparities across dialects (27.7-59.4 WER range), which cannot be resolved through equal data distribution alone. Promising directions including cross-dialectal transfer learning and extending voice conversion techniques to ASR tasks. Our competitive rankings despite using only shared task data demonstrate that advancing dialec-

System	ALG	EGY	JOR	MAU	MOR	PAL	UAE	YEM	AVG
Whisper (Zero-shot)	79.58	81.37	19.28	82.89	80.42	77.92	80.27	80.58	84.69
BYZÖ-ctc (Zero-shot)	32.65	16.68	11.23	39.47	28.52	16.07	12.76	18.23	21.95
BYZÖ-ctc + SFT	**20.17**	13.06	12.25	**24.64**	16.20	13.91	11.68	**15.32**	15.90
BYZÖ-ctc + SFT + LM	22.03	12.02	10.17	26.25	15.89	12.30	11.00	15.85	15.69
BYZÖ-whisper + SFT I	26.69	13.41	10.36	30.12	21.21	12.23	11.91	24.79	18.84
BYZÖ-whisper + SFT II	20.59	**11.91**	**9.47**	24.85	**15.52**	**10.59**	**9.04**	16.05	**14.76**

Table 3: Character Error Rate (CER) performance across eight Arabic dialects on the NADI 2025 test set. All our systems used only shared task data for dialect adaptation. The baseline Whisper zero-shot results demonstrate the challenge of dialectal ASR, while our BYZÖ systems show progressive improvements. Best result for a dialect is shown in bold. Lower values indicate better performance.

tal Arabic speech technology requires not massive resources, but careful adaptation strategies tailored to the unique characteristics of Arabic dialects.

Acknowledgments

This research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project-ID 232722074 – SFB 1102 and by the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the project TRAILS (01IW24005).

References

Ahmed Amine Ben Abdallah, Ata Kabboudi, Amir Kanoun, and Salah Zaiem. 2024. Leveraging data collection and unsupervised learning for code-switched tunisian arabic automatic speech recognition. In *ICASSP 2024-2024 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)*, pages 12607–12611. IEEE.

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, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. Nadi 2023: The fourth nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2310.16117*.

Muhammad Abdul-Mageed, Amr Keleg, AbdelRahim Elmadany, Chiyu Zhang, Injy Hamed, Walid Magdy, Houda Bouamor, and Nizar Habash. 2024. Nadi 2024: The fifth nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2407.04910*.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. NADI 2020: The first nuanced arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop, WANLP@COLING 2020, Barcelona, Spain (Online), December 12, 2020*, pages 97–110. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. Nadi 2022: The third nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2210.09582*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim A. 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, WANLP 2021, Kyiv, Ukraine (Virtual), April 9, 2021*, pages 244–259. Association for Computational Linguistics.

Badr M Abdullah, Matthew Baas, Bernd Möbius, and Dietrich Klakow. 2025. Voice conversion improves cross-domain robustness for spoken arabic dialect identification. *arXiv e-prints*, pages arXiv–2505.

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith Abandah, Adham Alsharkawi, and Maha Dawas. 2023. Masc: Massive arabic speech corpus. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 1006–1013.

Karim Al-Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a Levantine corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 769–778, Marseille, France. European Language Resources Association.

Sadeen Alharbi, Areeb Alowisheq, Zoltán Tüske, Kareem Darwish, Abdullah Alrajeh, Abdulmajeed Alrowithi, Aljawharah Bin Tamran, Asma Ibrahim, Raghad Aloraini, Raneem Alnajim, and 1 others. 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.

Anass Allak, Naira Abdou Mohamed, Imade Benelallam, and Kamel Gaanoun. 2021. Dialectal voice : An open-source voice dataset and automatic speech recognition model for moroccan arabic dialect. In *Proceedings of the Data Centric AI (NeurIPS 2021)*.

R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215.

Matthew Baas, Benjamin van Niekerk, and Herman Kamper. 2023. Voice conversion with just nearest neighbors. In *Interspeech 2023*, pages 2053–2057.

Omar A Essameldin, Ali O Elbeih, Wael H Gomaa, and Wael F Elsersy. 2025. Arabic dialect classification using rnns, transformers, and large language models: A comparative analysis. *arXiv preprint arXiv:2506.19753*.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Clive Holes. 2004. *Modern Arabic: Structures, functions, and varieties*. Georgetown University Press.

Mustafa Jarrar, Fadi A Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2022. Lisan: Yemeni, iraqi, libyan, and sudanese arabic dialect copora with morphological annotations.

Abdullah Khered, Ingy Abdelhalim, Nadine Abdelhalim, Ahmed Soliman, and Riza Batista-Navarro. 2023. UniManc at NADI 2023 shared task: A comparison of various t5-based models for translating Arabic dialectical text to Modern Standard Arabic. In *Proceedings of ArabicNLP 2023*, pages 658–664, Singapore (Hybrid). Association for Computational Linguistics.

Yunpeng Liu, Xukui Yang, and Dan Qu. 2024. Exploration of whisper fine-tuning strategies for low-resource asr. *EURASIP Journal on Audio, Speech, and Music Processing*, 2024(1):29.

Hedi Naouara, Jean-Pierre Lorré, and Jérôme Louradour. 2025. Linto audio and textual datasets to train and evaluate automatic speech recognition in tunisian arabic dialect. *arXiv preprint arXiv:2504.02604*.

Ömer Tarik Özyilmaz, Matt Coler, and Matias Valdenegro-Toro. 2025. Overcoming data scarcity in multi-dialectal arabic asr via whisper fine-tuning. *arXiv preprint arXiv:2506.02627*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Peter Sullivan, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. On the robustness of arabic speech dialect identification. In *Interspeech 2023*, pages 5326–5330.

Bashar Talafha, Karima Kadaoui, Samar Mohamed Magdy, Mariem Habiboullah, Chafei Mohamed Chafei, Ahmed Oumar El-Shangiti, Hiba Zayed, Rahaf Alhamouri, Rwaa Assi, Aisha Alraeesi, and 1 others. 2024. Casablanca: Data and models for multidialectal arabic speech recognition. *arXiv preprint arXiv:2410.04527*.

Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarar, Nizar Habash, and Muhammad Abdul-Mageed. 2025. Nadi 2025: The first multidialectal arabic speech processing shared task. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

A Training parameters for Whisper

A.1 System Configurations and Training Setup

We train and evaluate three Whisper-based ASR systems, summarized as follows:

1. **Whisper + SFT**: A two-stage fine-tuning system with WER loss in first and second stage. This configuration trains all model weights (no LoRA). The maximum generation length used in training is 225 tokens.

2. **Whisper + SFT + 2OPT**: A full fine-tuning system with WER loss in first stage (same shared across all systems) and CER loss for the second stage. This configuration trains all model weights (no LoRA) to directly compare against System **Whisper + SFT** and show the effect of CER-based training. The maximum generation length is 225 tokens.

3. **Whisper + SFT + LORA + 2OPt**: We use the same first stage model trained using full fine-tuning system with WER loss and for the scond stage we train a parameter-efficient system using LoRA and a CER loss. because it showed that it was effective in System **Whisper + SFT + 2OPT**. We freeze Whisper's original weights and fine-tune only LoRA adapter parameters inserted in each layer (rank $r = 32$). The CER loss term ($\lambda = 0.5$) is added to the training objective to directly optimize character accuracy. We impose a stricter maximum generation length of 125 tokens to

simulate potential truncation and evaluate its effect, especially in conjunction with LoRA. This setup updates only ∼1% of parameters, significantly reducing training memory and time, making it appealing for low-resource or deployment scenarios if accuracy trade-offs are acceptable.

B Training Corpora of n-gram Language Models

The training corpora for the n-gram language models were compiled from several existing, dialect-annotated datasets. These primary sources include the Palestinian Curas corpus (Al-Haff et al., 2022), the Yemeni Lisan corpus (Jarrar et al., 2022), the Emirati Emi-NADI (Khered et al., 2023), the Moroccan Darija-LID dataset[2], and the multi-dialect QADI corpus (Abdelali et al., 2021).

To augment these resources, we expanded the training data by automatically annotating a subset of the Arabic-tweets dataset (Al-Fetyani et al., 2023). This dialect identification task was performed using the MARBERTv2 model (Essameldin et al., 2025).

C Correlation between Different Models

Figure 1 shows the correlation between different models in their dialect performance. One can observe a strong correlation between the models, which indicates that the different systems behave similarly for the dialectal Arabic ASR task.

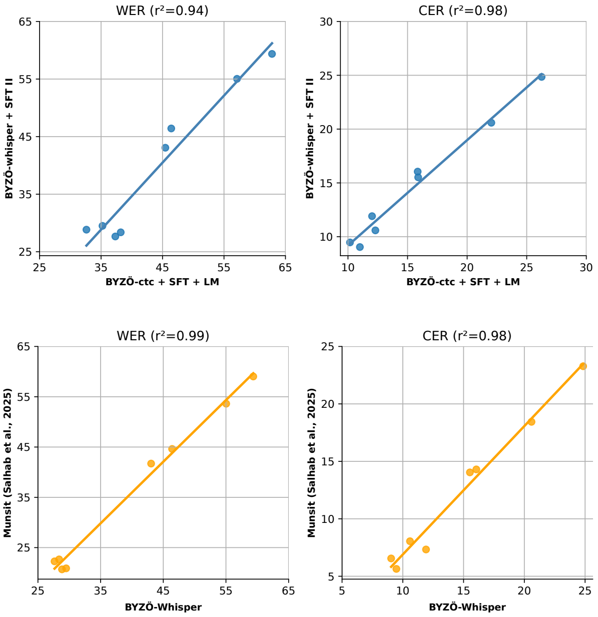


Figure 1: Performance correlation between different models: Our CTC- and Whisper-based systems (top), and the top performing system in the shared task vs. our best system. Each data point in the figure corresponds to a dialect.

[2] https://huggingface.co/datasets/atlasia/Darija-LID

MarsadLab at NADI Shared Task: Arabic Dialect Identification and Speech Recognition using ECAPA-TDNN and Whisper

Md. Rafiul Biswas[1], Kais Attia[2], Shimaa Ibrahim[3],
Mabrouka Bessghaier[3], **Wajdi Zaghouani**[3]
[1]Hamad Bin Khalifa University, Qatar, [2]Independent Researcher, Tunisia
[3]Northwestern University in Qatar, Qatar
mbiswas@hbku.edu.qa,wajdi.zaghouani@northwestern.edu

Abstract

We participated in NADI 2025 shared tasks on Arabic Dialect Identification (ADI) and Automatic Speech Recognition (ASR) across eight Arabic dialects. For ADI, we employ an enhanced ECAPA-TDNN with VoxLingua107 initialization, featuring self-attention classification head, progressive unfreezing, advanced augmentation, and test-time augmentation. This approach ranked third with 61.6% accuracy and 0.3068 macro cost. For ASR, we implement a zero-shot cascaded system using Whisper Large-v3 and MARBERT with extreme parameter efficiency (0.0004% trainable), ranking seventh with 104.895 WER and 84.693 CER. Our results validate complementary paradigms: direct audio processing for competitive dialect classification versus foundation model robustness for cross-dialectal transcription.

1 Introduction

Arabic is a pluricentric language with a rich continuum of regional and social varieties. This diversity—spanning Egyptian, Levantine, Gulf, Maghrebi, and other dialect groupings alongside Modern Standard Arabic (MSA)—poses unique challenges for speech technologies (Rahman et al., 2024). Despite steady progress in speech processing, reliable recognition and identification of Arabic dialects from speech remains difficult due to limited labeled resources, frequent code-switching with MSA and other languages, and substantial phonetic and lexical variation (Biadsy et al., 2009). Earlier shared tasks on spoken dialect identification helped define the problem space and catalyze benchmarking (Ali et al., 2017, 2019) while recent large-scale models that jointly learn ASR and language identification—such as Whisper (Tang et al., 2022; Radford et al., 2022) and MMS (Pratap et al., 2023) have reset expectations for zero-/few-shot performance. Still, their effectiveness on multidialectal Arabic, especially under domain shift and fine-grained dialect labels, is far from settled (Aboelela and Mansour, 2025).

The NADI 2025 shared task (Talafha et al., 2025) addresses two complementary problems: fine-grained dialect identification from single utterances and robust ASR across dialects using the Casablanca dataset. Building on prior Arabic shared tasks and benchmarks (e.g., MGB-3 (Ali et al., 2017), MGB-2 (Ali et al., 2019)), we adopt two complementary system designs: (1) an adaptation-heavy ECAPA-TDNN (Desplanques et al., 2020a) pipeline for dialect classification and (2) a zero-shot Whisper Large baseline for ASR. Our design choices emphasize reproducibility and computational practicality while exploring methods that improve dialect discrimination and transcription robustness. Our proposed system model using Whisper and MMS dataset demonstrates the power of large-scale multilingual models. community-driven effort to advance multidialectal Arabic speech recognition, while Speechbrain (Ravanelli et al., 2021a), VoxLingua107 (Valk and Alumäe, 2021) and ECAPA-TDNN (Desplanques et al., 2020a) provide crucial multilingual and architectural foundations.

Our contributions are threefold: (i) a practical and reproducible Arabic ASR that is based on ECAPA TDNN that features the self-attention mechanism; (ii) an empirical study of the use of OpenAI Whisper Large v3 in Casablanca for dialect-specific transcription; and (iii) a transparent analysis of errors and per-dialect behavior to inform future multidialectal modeling.

2 Background

NADI subtasks uses Casablanca audio corpus covering eight target dialects (Algerian, Egyptian, Jordanian, Mauritanian, Moroccan, Palestinian, Emirati, Yemeni)(Talafha et al., 2024). Each input is a single-channel WAV file carrying one utter-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 752–756
November 8-9, 2025 ©2025 Association for Computational Linguistics

ance; ADI expects a single dialect label output and ASR expects a text transcription (MSA or dialectal Arabic depending on the speaker). Table 1 presents the NADI 2025 Arabic dialect dataset comprising 25,600 audio samples across 8 Arabic dialects. The dataset is well-balanced with each dialect containing exactly 3,200 samples, split nearly evenly between training (12,900) and validation (12,700) sets. Audio recordings are sampled at 16 kHz with durations ranging from 1.04 to 15.12 seconds (mean: 4.25s, median: 3.56s, std: 2.79s). An additional 6,268 unlabeled test samples are provided for evaluation.

Figure 1 illustrates the audio characteristics analysis of the dataset. The left panel shows the distribution of audio durations, revealing a right-skewed distribution with most samples concentrated between 2-4 seconds, and the mean (4.3s) slightly higher than the median (3.6s) due to longer outliers. Dialects exhibit similar interquartile ranges and median values around 3-4 seconds. Both visualizations confirm the dataset's consistency and balance, making it suitable for robust Arabic dialect identification model training and evaluation.

Dialect	**Train**	**Val**
Algeria	1,610	1,590
Egypt	1,603	1,597
Jordan	1,604	1,596
Mauritania	1,617	1,583
Morocco	1,608	1,592
Palestine	1,631	1,569
UAE	1,602	1,598
Yemen	1,625	1,575
Total	**12,900**	**12,700**
Dataset Overview		
Dialects: 8	Total: 25,600	Test: 6,268
Sampling rate: 16 kHz		
Audio Duration Statistics (seconds)		
Mean: 4.25	Median: 3.56	Std: 2.79
Range: 1.04 – 15.12		

Table 1: Dialectal distribution of NADI dataset

Task Challenges: Prior Arabic speech work demonstrates recurring challenges: dialectal variation, scarcity of labeled data for many dialects, and domain mismatch between broadcast and in-the-wild audio (Ali et al., 2017; Althobaiti, 2020). Recent multilingual foundation models (Whisper (Radford et al., 2022), MMS (Pratap et al., 2023)) show strong zero-shot generalization, while architectures such as ECAPA-TDNN have been effective for representation extraction in speaker and language tasks (Desplanques et al., 2020b). For implementation and tooling we relied on the SpeechBrain toolkit (Ravanelli et al., 2021b).

3 System Overview

We implemented two systems consistent with the memorized process described earlier. Below we summarize the main design choices and components for each subtask.

3.1 Subtask 1: Dialect Identification (ECAPA-TDNN pipeline)

Base architecture: ECAPA-TDNN pre-trained and described in prior work (Desplanques et al., 2020b). We adapt ECAPA as a robust embedding extractor and add a classification pathway on top.

Classification head: Custom multi-layer MLP with Swish activation, BatchNorm, dropout, and a feature-wise self-attention module. The attention reweights ECAPA feature vectors:

$$\begin{aligned} \mathbf{a} &= \sigma(\mathbf{W}_2 \cdot \mathsf{Swish}(\mathbf{W}_1\mathbf{h} + \mathbf{b}_1) + \mathbf{b}_2)\,, \\ \hat{\mathbf{h}} &= \mathbf{a} \odot \mathbf{h}, \end{aligned} \tag{1}$$

Training schedule:

- Phase 1: Freeze ECAPA backbone; train classifier head (2,500 steps).
- Phase 2: Unfreeze top ECAPA layers; fine-tune with discriminative learning rates ($\eta_{\text{encoder}} = 1 \times 10^{-6}, \eta_{\text{classifier}} = 5 \times 10^{-5}$).

Loss & regularization: Combined loss $\mathcal{L} = 0.3\mathcal{L}_{\text{focal}} + 0.7\mathcal{L}_{\text{CE}}$ (focal $\gamma = 2.5$), label smoothing, gradient clipping, and cosine-annealing LR with warmup.

Augmentation & inference: Advanced augmentation pipeline (noise, pitch/time perturbations, reverb, volume, frequency/time masking) during training. At inference we applied Test-Time Augmentation (TTA) with 5–10 variants per utterance and averaged softmax outputs; temperature scaling was used for calibration.

3.2 Subtask 2: Automatic Speech Recognition (MARBERT-Whisper pipeline)

We employ OpenAI Whisper Large-v3 (via Hugging Face `pipeline("automatic-speech-recognition")`) as our baseline (Radford et al., 2022). Our approach implements a cascaded architecture for Arabic Dialect Identification (ADI), combining ASR with text classification through parameter-efficient transfer learning.

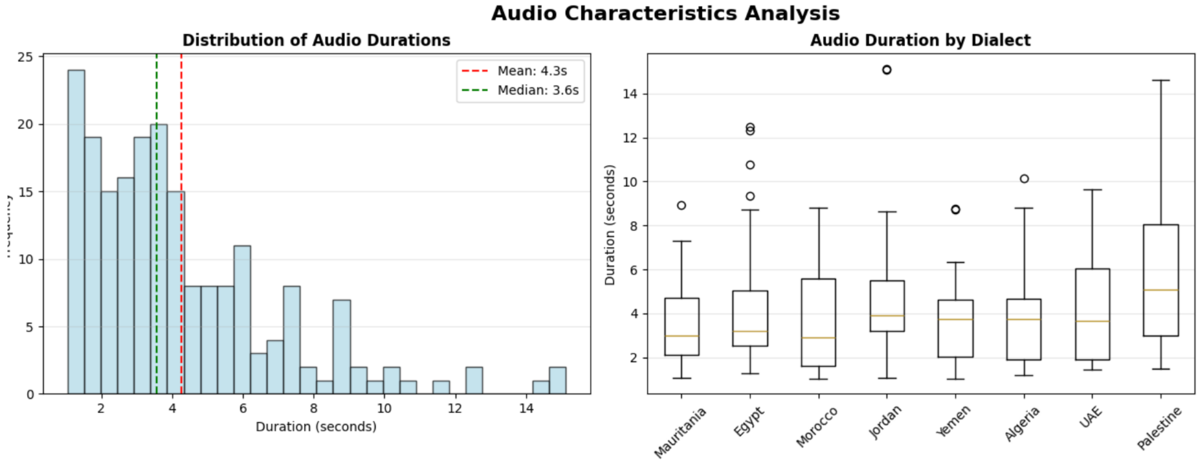


Figure 1: Statistical distribution of audio duration in NADI dataset

Given an input audio signal $\mathbf{x} \in \mathbb{R}^T$ of length T, the system performs sequential transformations:

1. **ASR:** Whisper Large-v3 for speech-to-text.
2. **Encoding:** MARBERT for contextual text embeddings.
3. **Classification:** Trainable linear layer for dialect prediction.

The speech-to-text step uses a frozen Whisper model Φ_{whisper}:

$$t = \Phi_{\text{whisper}}(\mathbf{x}; \boldsymbol{\theta}_{\text{whisper}}), \tag{2}$$

where t is the transcript and $\boldsymbol{\theta}_{\text{whisper}}$ are frozen pretrained parameters.

The transcript t is processed by the frozen MARBERT encoder Φ_{MARBERT}:

$$\mathbf{h} = \Phi_{\text{MARBERT}}(t; \boldsymbol{\theta}_{\text{MARBERT}}), \tag{3}$$

where $\mathbf{h} \in \mathbb{R}^{768}$ is the [CLS] token embedding.

A trainable classifier maps $\mathbf{h}$ to dialect probabilities:

$$\mathbf{y} = \text{softmax}(\mathbf{W}\mathbf{h} + \mathbf{b}), \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{8\times768}$ and $\mathbf{b} \in \mathbb{R}^8$ are the only trainable parameters.

Audio is resampled to 16 kHz mono, truncated at 30 s, and zero-padded. Text is tokenized with MARBERT (max length 512, dynamic padding). Training uses batch-mode transcript processing; inference is sequential with error handling.

This cascaded design achieves $\mathcal{O}(T \log T)$ complexity for ASR and $\mathcal{O}(L^2)$ for encoding, with minimal overhead due to selective parameter updates.

Component	Task 1: ADI	Task 2: ASR
Framework	SpeechBrain	HF Transformers
Pretrained	ECAPA-TDNN	Whisper Large + MARBERT
Optimizer	AdamW	AdamW
Batch size	32	8 (train), 4 (val)
Precision	FP16	FP16
Augmentation	Audio perturb.	None
Learning rate	5e-5	2e-5
Trainable params	Enhanced classifier	6,152 (0.0004%)
Max steps	25,000	3,000
Hardware	8 GB+ GPU	8 GB+ GPU

Table 2: Training configurations for ADI and ASR tasks

4 Experimental Setup

All experiments used the organizer-provided splits (Table 1). Implementations used SpeechBrain for ECAPA-based pipelines and Hugging Face Transformers for Whisper. Important implementation details are summarized in Table 2.

Metrics and evaluation. For ADI we report accuracy and the macro-averaged cost metric provided by the organizers. For ASR we report average WER and CER using the Codabench evaluation script. Recent large-scale approaches and multilingual systems motivate the use of zero-shot baselines for comparison (Pratap et al., 2023; Radford et al., 2022).

5 Results

Table 3 summarizes official results submitted to the organizers and used for official ranking. The enhanced ECAPA-TDNN system achieved a competition score of 0.616 (cost: 0.3068) in Task 1, demonstrating competitive performance against the best system which scored 0.7983 (cost: 0.1788),

Task	Metric 1	Metric 2	Rank
ADI	Acc. 0.616	Macro Cost 0.3068	3
ASR	Avg. WER 104.90	Avg. CER 84.69	7

Table 3: Performance metrics of our proposed system

validating the effectiveness of direct audio processing for Arabic dialect identification.

For Task2, the novel Whisper + MARBERT cascaded approach, while achieving more modest accuracy, offers significant advantages in computational efficiency and interpretability, requiring only minimal parameter training while leveraging the power of large pre-trained models.

5.1 Ablation and analysis (validation splits)

We performed ablations during development on the validation set. Removing the feature-wise attention layer reduced validation discrimination between similar dialect classes and led to decreased stability in low-resource dialects (consistent with our informal validation runs). Progressive unfreezing and discriminative learning rates helped preserve pretrained representations and improved final validation cost.

5.2 Error analysis

We analyzed common confusions on validation and test samples (explicitly noting which split is used where):

- **Dialect confusions:** Moroccan and Algerian Arabic sound very similar in how they're spoken (rhythm/melody) and use similar words/expressions. The same applies to Levantine and Palestinian Arabic. When these linguistic features "overlapped" (were very similar between the pairs), the AI system couldn't reliably distinguish between them.

- **ASR errors:** Whisper zero-shot produced frequent errors in colloquial and code-switched segments (e.g., mixing Arabic and French terms), and often omitted short function words or mis-transcribed named entities.

Example (validation): a Moroccan utterance containing dialectal lexical items was misclassified as Algerian due to shared lexical forms and similar rhythm; manual inspection revealed low SNR and overlapped background speech.

6 Discussion

Our NADI 2025 participation reveals several critical limitations and areas for improvement across both tasks. For Task 1 (ADI), our enhanced ECAPA-TDNN system achieved an accuracy of 0.616 with macro cost of 0.3068, ranking 3rd among participants, compared to the best performing system at 0.7983, indicating substantial room for optimization in fine-tuning strategies and feature extraction despite our sophisticated enhancement techniques including self-attention mechanisms, progressive unfreezing, and advanced data augmentation.

The cascaded approach in Task 2 (ASR) exposed fundamental limitations of speech-to-text pipelines, achieving an average WER of 104.90 and CER of 84.69, ranking 7th in the competition. These high error rates reflect domain mismatch between Whisper's training data and the competition dataset, as well as differences in transcription conventions and dialectal variations that the pre-trained model was not optimized for. Error propagation from the ASR component directly impacts downstream classification performance, as dialectal acoustic features crucial for identification are lost during transcription. This suggests that preserving prosodic and phonetic information through direct audio processing remains superior for dialect-specific tasks.

The limited training data for certain dialect classes exacerbated class imbalance issues in Task 1, despite employing focal loss and data augmentation techniques, while the extremely high error rates in Task 2 suggest fundamental challenges in adapting general-purpose ASR models to dialectal Arabic. Future improvements should focus on dialectal data augmentation strategies, cross-lingual transfer learning from related Arabic varieties, hybrid architectures that combine acoustic and linguistic features for ADI, and specialized ASR models trained specifically on dialectal Arabic corpora.

7 Conclusion

In summary, our experiments presents the complementary strengths of two paradigms: fine-tuned ECAPA-TDNN, augmented with diverse perturbations and targeted architectural refinements, delivers strong dialect classification, whereas Whisper Large serves as a capable zero-shot transcription baseline across dialects without any task-specific adaptation. This contrast suggests a promising avenue in combining the adaptability of tailored acoustic models with the broad coverage of large, general-purpose ASR systems.

Code Reproducibility

To ensure reproducibility of our results, all source code, model implementations, and experimental configurations are made publicly available at `https://github.com/rafiulbiswas/NADI`. The repository includes complete implementations for both tasks with detailed documentation and setup instructions.

Acknowledgments

This study was supported by the grant NPRP14C-0916-210015, awarded by the Qatar Research, Development and Innovation Council (QRDI).

References

Eman Aboelela and Omar Mansour. 2025. A review of speech recognition and application to arabic speech recognition. In *Future of Information and Communication Conference*, pages 13–31. Springer.

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2019. The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. *Preprint*, arXiv:1609.05625.

Ahmed Ali, Stephan Vogel, and Steve Renals. 2017. Speech recognition challenge in the wild: Arabic mgb-3. *Preprint*, arXiv:1709.07276.

Maha J. Althobaiti. 2020. Automatic arabic dialect identification systems for written texts: A survey. *Preprint*, arXiv:2009.12622.

Fadi Biadsy, Julia Bell Hirschberg, and Nizar Y Habash. 2009. Spoken arabic dialect identification using phonotactic modeling.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020a. ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech 2020*, pages 3830–3834.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020b. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. In *Interspeech 2020*, pages 3830–3834.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. Scaling speech technology to 1,000+ languages. *Preprint*, arXiv:2305.13516.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Ashifur Rahman, Md Mohsin Kabir, Muhammad Firoz Mridha, Mohammed Alatiyyah, Haifa F Alhasson, and Shuaa S Alharbi. 2024. Arabic speech recognition: Advancement and challenges. *IEEE Access*, 12:39689–39716.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, and 2 others. 2021a. SpeechBrain: A general-purpose speech toolkit. *Preprint*, arXiv:2106.04624. ArXiv:2106.04624.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, and 2 others. 2021b. Speechbrain: A general-purpose speech toolkit. *Preprint*, arXiv:2106.04624.

Bashar Talafha, Karima Kadaoui, Samar Mohamed Magdy, Mariem Habiboullah, Chafei Mohamed Chafei, Ahmed Oumar El-Shangiti, Hiba Zayed, Rahaf Alhamouri, Rwaa Assi, Aisha Alraeesi, and 1 others. 2024. Casablanca: Data and models for multidialectal arabic speech recognition. *arXiv preprint arXiv:2410.04527*.

Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarar, Nizar Habash, and Muhammad Abdul-Mageed. 2025. Nadi 2025: The first multidialectal arabic speech processing shared task. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Raphael Tang, Karun Kumar, Gefei Yang, Akshat Pandey, Yajie Mao, Vladislav Belyaev, Madhuri Emmadi, Craig Murray, Ferhan Ture, and Jimmy Lin. 2022. Speechnet: Weakly supervised, end-to-end speech recognition at industrial scale. *arXiv preprint arXiv:2211.11740*.

Jörgen Valk and Tanel Alumäe. 2021. VoxLingua107: a dataset for spoken language recognition. In *Proc. IEEE SLT Workshop*.

Abjad AI at NADI 2025: CATT-Whisper: Multimodal Diacritic Restoration Using Text and Speech Representations

Ahmad Ghannam, Naif Alharthi, Faris Alasmary, Kholood Al Tabash,
Shouq Sadah, and Lahouari Ghouti
Abjad AI. King Khaled Road. Riyadh 11000, Saudi Arabia.
{aghannam,nalharthi,falasmary,kaltabash,ssadah,lghouti}@abjad.com.sa

Abstract

In this work, we tackle the Diacritic Restoration (DR) task for Arabic dialectal sentences using a multimodal approach that combines both textual and speech information. We propose a model that represents the text modality using an encoder extracted from our own pre-trained model named CATT. The speech component is handled by the encoder module of the OpenAI Whisper base model. Our solution is designed following two integration strategies. The former consists of fusing the speech tokens with the input at an early stage, where the 1500 frames of the audio segment are averaged over 10 consecutive frames, resulting in 150 speech tokens. To ensure embedding compatibility, these averaged tokens are processed through a linear projection layer prior to merging them with the text tokens. Contextual encoding is guaranteed by the CATT encoder module. The latter strategy relies on cross-attention, where text and speech embeddings are fused. The cross-attention output is then fed to the CATT classification head for token-level diacritic prediction. To further improve model robustness, we randomly deactivate the speech input during training, allowing the model to perform well with or without speech. Our experiments show that the proposed approach achieves a word error rate (WER) of 0.25 and a character error rate (CER) of 0.9 on the development set. On the test set, our model achieved WER and CER scores of 0.55 and 0.13, respectively.

1 Introduction

Diacritics are essential for accurate interpretation, pronunciation, and meaning in Arabic. However, in most informal writing such as social media, messaging, or transcribed speech they are omitted. While native speakers often infer the intended forms from context, the absence of diacritics introduces significant ambiguity, particularly in dialects where phonetic and morphological variation is high and orthographic conventions are inconsistent. This not only challenges human readers but also degrades the performance of downstream NLP tasks such as speech synthesis, machine translation, and information retrieval. The NADI 2025 shared task overview (Talafha et al., 2025) highlights that DR remains particularly difficult for dialectal Arabic due to limited annotated data, regional variability, and inconsistent spelling practices. Traditional DR approaches rely solely on text, ranging from rule-based systems and n-gram models to transformer-based language models such as BERT (Devlin et al., 2019). These methods often fail when orthographic cues alone are insufficient, an issue exacerbated in dialectal and code-switched text. In contrast, speech carries prosodic and phonetic signals that can directly disambiguate diacritic placement, offering a valuable complement to text.

In this work, we propose CATT-Whisper, a multimodal DR system that integrates a CATT (Alasmary et al., 2024) text encoder with the Whisper (Radford et al., 2023) speech encoder. We evaluated two fusion strategies: **(i) Early fusion**: projected speech embeddings are merged with text embeddings before passing them to CATT encoder as inputs. **(ii) Cross-attention fusion**: the output of the CATT encoder is fused with the speech embeddings from Whisper using cross attention layer, followed by the classification layer.

Our contributions are: (i) A multimodal DR system for Arabic dialects combining large-scale pre-trained text and speech encoders. (ii) Comparative analysis of early fusion vs. Cross-attention fusion. (iii) A modality-robust training scheme for variable speech availability. Our full codebase, including pre-trained models and training scripts, is publicly available [1], ensuring reproducibility and facilitating further research in multimodal DR.

[1] https://github.com/abjadai/catt-whisper

Proceedings of The Third Arabic Natural Language Processing Conference, pages 757–761
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Background

2.1 Task Setup

The DR shared subtask at NADI 2025 focuses on restoring missing diacritics in Arabic text, with the option to also use speech for better performance. Unlike most previous work that only targets MSA, this task also covers Classical Arabic, dialects, and code-switched text, which are more challenging. Some examples are provided in Table 1.

Example Input 1	عندكو شوربة ايه النهرده
CATT	عِنْدَكُو شُوْرْبَةُ ايه النَّهْرَدَهْ
CATT-Whisper	عَنْدُكُو شورِبة اِيه النِهَردَه
Reference	عَنْدُكُو شورِبة اِيه النِهَردَه
Example Input 2	عايز شوية وأت لتجهيز الاكل
CATT	عَايَزَ شُوِيَّةً وَأْتْ لِتَجْهِيزِ الاكْلِ
CATT-Whisper	عَايِز شوَيَّة وَأت لِتَجهِيز الاَكل
Reference	عَايِز شوَيَّة وَأت لِتَجهِيز الاَكل

Table 1: Examples from the NADI 2025 Subtask 3 dataset (dev/test). CATT (text-only) and CATT-Whisper (speech-enhanced) outputs compared with references, showing how speech features resolve phonological ambiguities.

2.2 Dataset

Our experiments were conducted using the NADI 2025 DR dataset, provided as part of the shared task, which is publicly available on Hugging Face [2]. The dataset covers a mix of dialectal, multi-dialectal, and Classical Arabic varieties, with some segments exhibiting code-switching between Arabic and other languages. The dataset is a combined collection derived from several resources, namely MDASPC (Almeman et al., 2013), TunSwitch (Abdallah et al., 2023), ArzEn (Hamed et al., 2020), Mixat (Al Ali and Aldarmaki, 2024), ClArTTS (Kulkarni et al., 2023), and ArVoice (Toyin et al., 2025). While the CATT and Whisper models we use in our system were already pretrained on their respective large-scale corpora, the NADI 2025 DR dataset used exclusively for fine-tuning the combined architecture for this DR task. The provided training data consists of multiple sub-datasets, summarized in Table 2.

Dataset	Type	Dia.	Train
MDASPC	Multi-dialectal	True	60,677
TunSwitch	Dialectal, CS	True	5,212
ArzEn	Dialectal, CS	False	3,344
Mixat	Dialectal, CS	False	3,721
ClArTTS	CA	True	9,500
ArVoice	MSA	True	2,507

Table 2: Statistics of the NADI 2025 Subtask 3 datasets. CA = Classical Arabic, CS = Code-Switched Arabic, Dia. = diacritic. The table reports the number of sentences in each split.

2.3 Related Work

Research on Arabic DR has evolved from rule-based methods to neural and multimodal approaches (Elgamal et al., 2024). Early systems relied on lexicons and morphological analyzers, later extended with n-gram models (Habash and Rambow, 2007; Elshafei et al., 2006), but they struggled with dialectal variation, noisy text, and borrowed vocabulary. Neural models, from RNNs and LSTMs (Zitouni et al., 2006; Belinkov and Glass, 2015) to transformers (Nazih and Hifny, 2022) with pre-trained language models such as AraBERT (Antoun et al., 2020), CAMeLBERT (Inoue et al., 2021), and CATT (Alasmary et al., 2024), improved accuracy but still failed to resolve phonetic ambiguities in dialects. While (Elgamal et al., 2024) highlighted the usefulness of "diacritics-in-the-wild" signals, text-only models remain insufficient for ambiguous cases.

Multimodal approaches increasingly exploit ASR outputs as phonetic cues. Early work (Aldarmaki and Ghannam, 2023) relies solely on ASR, which can produce both transcripts and diacritic predictions, but errors in transcription often propagate to diacritization. More recent methods (Shatnawi et al., 2024a) integrate ASR-derived-diacritized transcripts with undiacritized text via cross-attention, enhancing performance while still being sensitive to ASR noise.

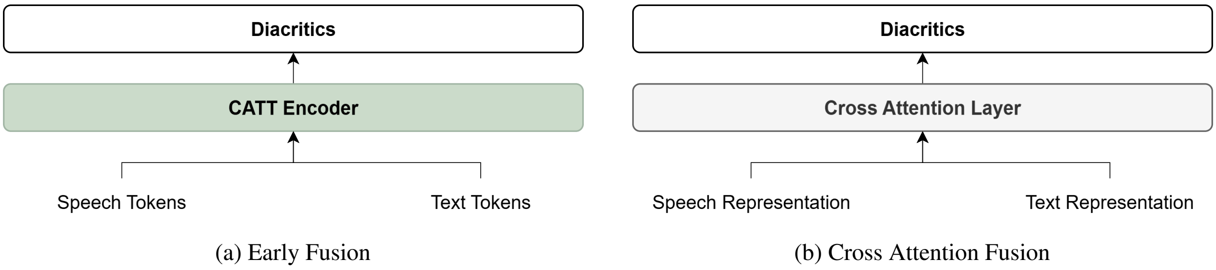
Our approach differs by (i) deeply integrating text and speech through early and cross-attention fusion, (ii) focusing explicitly on dialectal DR with robust pre-trained encoders: CATT and Whisper.

3 System Overview

3.1 Architecture Components

The architecture consists of a **Text Encoder**, implemented with a pre-trained CATT model for DR, and

[2] https://huggingface.co/datasets/MBZUAI/NADI-2025-Sub-task-3-all



(a) Early Fusion

(b) Cross Attention Fusion

Figure 1: Proposed CATT-Whisper Architectures for Multimodal. (a) Early Fusion Configuration. (b) Cross-Attention Fusion Configuration.

a **Speech Encoder**, implemented with the Encoder part of Whisper-Base model. A **Linear Projection Layer** follows the speech encoder to match the dimensionality of the text encoder. The proposed architectures are summarized in Figure 1.

3.2 Fusion Strategies

3.2.1 Early Fusion

Speech features are downsampled from 1,500 frames to 150 tokens by averaging 10 frames with and projecting them to match the text embedding dimension. These speech tokens are then concatenated with text tokens and fed into the CATT encoder, following a strategy similar to (Wu et al., 2023). This early fusion approach can be seen as a form of "soft prompting," where text tokens are augmented with speech embeddings via speech-placeholder tokens, enabling the model to leverage acoustic features while preserving the core CATT architecture. Details of this fusion strategy is shown in Figure 2.

3.2.2 Cross-Attention Fusion

Text and speech embeddings are encoded separately, then fused via a cross-attention layer before being passed to the classification layer, similar to the multi-modal setup of (Shatnawi et al., 2024b).

3.2.3 Fusion Strategy Choice

In our experiments, both Early Fusion and Cross-Attention Fusion yielded comparable results. However, as Cross-Attention is computationally more demanding, we focused on Early Fusion, and all results reported in this paper correspond to this configuration.

3.3 Speech Augmentation

Time-frequency warping (Park et al., 2019) is applied during training to improve generalization.

4 Experimental Setup

For training, we used the NADI 2025 DR train and development sets, while evaluation was performed on the official test set. Model performance was measured using Word Error Rate (WER) and Character Error Rate (CER), which are the standard metrics. Our preprocessing step included tokenization, speech feature extraction, and spectrogram augmentation through time-frequency warping.

Training was carried out with a batch size of 32, a learning rate of 1×10^{-5}, a dropout rate of 0.1, and the AdamW optimizer. During training, the speech encoder was frozen for the first 5 epochs allowing the projection layer to adapt, then unfrozen and jointly trained with the rest of the model for more 5 epochs. This two-phase procedure was applied in all experiments for both fusion models.

5 Results

5.1 Development Set Performance

Table 3 shows the performance of our proposed model compared to other works on the development set. Our model achieves substantially lower word error and character error rates (WER and CER).

Participant	WER	CER
gahmed92 (Ours)	**0.25**	**0.09**
omarnj	0.46	0.22
Baseline	0.46	0.22

Table 3: Results on the NADI 2025 Subtask 3 official development set, reported in WER and CER

5.2 Test Set Performance

Table 4 presents the results on the official test set. Our models outperforms all models in both metrics.

5.3 Performance on Challenging Test Cases

We further analyzed the model on a set of challenging test cases recorded by our team, where the

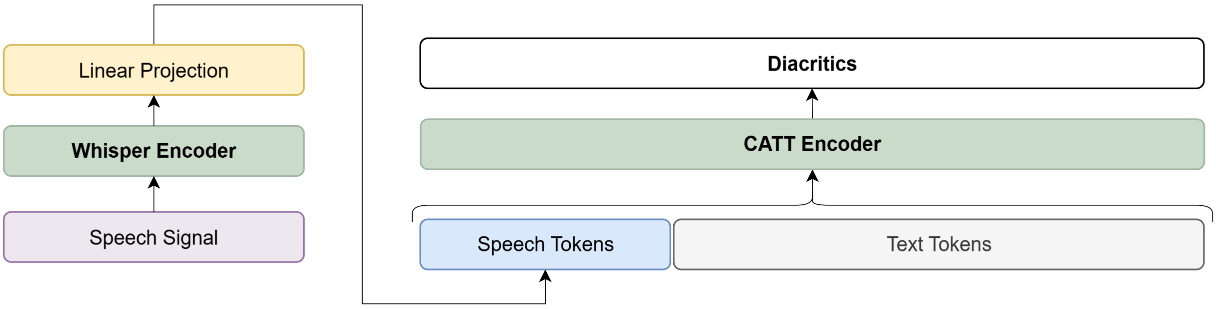


Figure 2: Early Fusion architecture of the proposed CATT-Whisper model. Speech features are downsampled and projected to match text embeddings before being concatenated with text tokens and processed by the CATT encoder.

Participant	WER	CER
gahmed92 (Ours)	**0.55**	**0.13**
mohamed_elrefai	0.64	0.15
Baseline	0.65	0.16

Table 4: Results on the NADI 2025 Subtask 3 official test set, reported in WER and CER

same word is pronounced differently within the same sentence. The results, summarized in Table 5, show that while our model achieves lower WER and CER than the others, these cases remain difficult and are not fully solved. This highlights both the robustness of our approach and the need for further improvements to handle complex, real-world pronunciation variability.

Example Input 1	ضرب ضرب ضرب
CATT-Whisper	ضَرِب ضَرِب ضَرِب
Reference	ضَرَبَ ضُرِبَ ضَرْبٌ
Example Input 2	ذهب ذهب
CATT-Whisper	ذَهِب ذَهِب
Reference	ذَهَبٌ ذَهَبْ

Table 5: Model performance on challenging test cases with variable word pronunciations.

6 Conclusion

We present CATT-Whisper, a multimodal system for Arabic DR that combines pre-trained text and speech encoders via early fusion and cross-attention. Both strategies achieve competitive results. While speech input boosts diacritic accuracy, some ambiguous sequences remain challenging, suggesting the need for stronger phoneme-level encoders (e.g., CTC-based models such as Conformer-CTC (Gulati et al., 2020), Squeezeformer (Kim et al., 2022)). Future work will explore alternative acoustic models and larger-scale training.

Acknowledgments

The authors would like to thank the management of AbjadAI Company for their generous support.

References

Ahmed Amine Ben Abdallah, Ata Kabboudi, Amir Kanoun, and Salah Zaiem. 2023. Leveraging data collection and unsupervised learning for code-switched tunisian arabic automatic speech recognition. *Preprint*, arXiv:2309.11327.

Maryam Khalifa Al Ali and Hanan Aldarmaki. 2024. Mixat: A data set of bilingual emirati-English speech. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 222–226, Torino, Italia. ELRA and ICCL.

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. In *Interspeech 2023*, pages 361–365.

Khalid Almeman, Mark Lee, and Ali Abdulrahman Almiman. 2013. Multi dialect arabic speech parallel corpora. In *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6.

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.

Yonatan Belinkov and James Glass. 2015. Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285, Lisbon, Portugal. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

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.

Moustafa Elshafei, Husni Al-Muhtaseb, and Mansour Alghamdi. 2006. Statistical methods for automatic diacritization of arabic text. *The Saudi 18th National Computer Conference. Riyadh*, 18:301–306.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020*, pages 5036–5040.

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, Rochester, New York. Association for Computational Linguistics.

Injy Hamed, Ngoc Thang Vu, and Slim Abdennadher. 2020. ArzEn: A speech corpus for code-switched Egyptian Arabic-English. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4237–4246, Marseille, France. European Language Resources Association.

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.

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. In *Advances in Neural Information Processing Systems*, volume 35, pages 9361–9373. Curran Associates, Inc.

Ajinkya Kulkarni, Atharva Kulkarni, Sara Abedalmon'em Mohammad Shatnawi, and Hanan Aldarmaki. 2023. Clartts: An open-source classical arabic text-to-speech corpus. In *2023 INTERSPEECH*, pages 5511–5515.

Waleed Nazih and Yasser Hifny. 2022. Arabic syntactic diacritics restoration using bert models. *Computational Intelligence and Neuroscience*, 2022.

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 *Proc. Interspeech 2019*, 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*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Sara Shatnawi, Sawsan Alqahtani, and Hanan Aldarmaki. 2024a. 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.

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.

Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarar, Nizar Habash, and Muhammad Abdul-Mageed. 2025. Nadi 2025: The first multidialectal arabic speech processing shared task. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Hawau Toyin, Rufael Marew, Humaid Alblooshi, Samar M. Magdy, and Hanan Aldarmaki. 2025. ArVoice: A Multi-Speaker Dataset for Arabic Speech Synthesis. In *Interspeech 2025*, pages 4808–4812.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, and Yu Wu. 2023. On decoder-only architecture for speech-to-text and large language model integration. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. 2006. Maximum entropy based restoration of Arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 577–584, Sydney, Australia. Association for Computational Linguistics.

ELYADATA & LIA at NADI 2025: ASR and ADI Subtasks

Haroun Elleuch[1,2,†], **Youssef Saidi**[1,‡], **Salima Mdhaffar**[2],
Yannick Estève[2], **Fethi Bougares**[1,2]

[1]ELYADATA, France [2]LIA, Avignon University, France

† Main contributor of the ADI subtask ‡ Main contributor of the ASR subtask
Correspondence: haroun.elleuch@elyadata.com

Abstract

This paper describes Elyadata & LIA's joint submission to the NADI multi-dialectal Arabic Speech Processing 2025. We participated in the Spoken Arabic Dialect Identification (ADI) and multi-dialectal Arabic ASR subtasks. Our submission ranked first for the ADI subtask and second for the multi-dialectal Arabic ASR subtask among all participants. Our ADI system is a fine-tuned Whisper-large-v3 encoder with data augmentation. This system obtained the highest ADI accuracy score of **79.83%** on the official test set. For multi-dialectal Arabic ASR, we fine-tuned SeamlessM4T-v2 Large (Egyptian variant) separately for each of the eight considered dialects. Overall, we obtained an average WER and CER of **38.54%** and **14.53%**, respectively, on the test set. Our results demonstrate the effectiveness of large pre-trained speech models with targeted fine-tuning for Arabic speech processing.

1 Introduction

Arabic is one of the most widely spoken languages in the world, both in terms of number of speakers and geographical spread (Lane, 2025). This wide distribution, coupled with centuries of contact with other languages and cultures, has led to the emergence of numerous colloquial varieties collectively known as Arabic dialects. Although the exact granularity and classification of these dialects remain a matter of debate, a common working assumption in computational processing is to associate a dialect with a country-level variety (Bouamor et al., 2014; Shon et al., 2020), or to a larger area where subdialects are the most similar (Gulf, Levant, North Africa) (Dhouib et al., 2022; Ali et al., 2017).

Dialectal Arabic poses unique challenges for speech and language processing. Unlike Modern Standard Arabic (MSA), dialects are predominantly spoken rather than written (Ferguson, 1959), with significant variation in phonology, lexicon, and syntax. They also lack standardized orthographic conventions, despite recent efforts such as CODA (Conventional Orthography for Dialectal Arabic) (Habash et al., 2012), and later efforts of Habash et al. (2018) and Alhafni et al. (2024). These properties complicate both Automatic Speech Recognition (ASR) and Automatic Dialect Identification (ADI) tasks, where systems must generalize across substantial linguistic variability.

The 2025 Nuanced Arabic Dialect Identification (NADI) Shared Task (Talafha et al., 2025) addresses these challenges through three subtasks aimed at improving the coverage and robustness of speech technologies for Arabic dialects:

- Spoken Arabic Dialect Identification (ADI)
- Multidialectal Arabic ASR using the recently released Casablanca dataset (Talafha et al., 2024)
- Diacritic Restoration focusing on dialectal variations of Arabic

Our team participated in the first two subtasks, achieving first place in ADI and second place in multi-dialectal ASR on the official test sets. In both cases, we leveraged large-scale pre-trained speech models with targeted fine-tuning strategies to address dialectal variability.

Our main contributions are (1) We propose an effective two-stage fine-tuning approach for ADI, using the Whisper-large-v3 encoder to achieve state-of-the-art results. (2) We demonstrate that separately fine-tuning the SeamlessM4T-v2 Large model for each dialect yields competitive ASR performances.

2 Arabic Dialect Identification

The ADI subtask aims to classify speech utterances into their respective country-level dialect categories automatically. Our approach leverages large-scale

Proceedings of The Third Arabic Natural Language Processing Conference, pages 762–766
November 8-9, 2025 ©2025 Association for Computational Linguistics

pre-trained speech representations and a two-stage fine-tuning process to effectively adapt to the dialectal nuances present in the provided dataset. In the following subsections, we describe the datasets used, our ADI model architecture, and the considered training strategy. We also present our experimental results and follow up with an analysis of the ADI system performances.

2.1 Datasets

We utilize several datasets to train and evaluate our ADI system, including established corpora covering multiple Arabic dialects, as well as the official NADI 2025 ADI dataset. The following is a detailed description of each dataset.

2.1.1 ADI-17 and ADI-20

The ADI-17 dataset (Shon et al., 2020) comprises 3,033 hours of dialectal Arabic speech from 17 country-level dialects for training, along with approximately 2 hours per dialect in the development and test splits, respectively.

The ADI-20 dataset (Elleuch et al., 2025) is an expanded and rebalanced version of ADI-17, extending its coverage from 17 to 20 Arabic varieties by including Tunisian and Bahraini dialects as well as Modern Standard Arabic (MSA). It also increases representation for previously underrepresented dialects, such as Jordanian and Sudanese, by incorporating additional speech material. In total, the training partition contains 3,556 hours of speech, while the development and test sets retain the same structure as in ADI-17, supplemented with approximately 2 hours per newly added variety in each split. To enable experiments under resource-constrained conditions and ensure per-dialect balance, ADI-20-53h, a stratified subset containing up to 53 hours of training data for each variety, resulting in a total of 1,060 hours is also available. Our future experiments will use this subset rather than the full ADI-20 dataset for the reasons mentioned earlier.

2.1.2 NADI 2025 ADI Dataset

The official dataset for the ADI subtask covers eight country-level Arabic dialects: Algeria, Egypt, Jordan, Mauritania, Morocco, Palestine, UAE, and Yemen. It includes an *adaptation* split of approximately 15 hours (12,900 utterances) with associated country labels, a validation split of similar size (12,700), and an eleven-hour held-out test set with 6268 utterances.

2.2 Model Architecture

Our system follows the best-performing configuration from Elleuch et al. (2025). The Whisper-large-v3 encoder (Radford et al., 2023) is used as a feature extractor, followed by an attention pooling layer that aggregates frame-level representations into fixed-length utterance embeddings. These are passed through a fully connected layer with a softmax activation for classification over the target dialects.

We freeze the first 16 layers of the Whisper encoder during fine-tuning to preserve general speech representations while adapting the upper layers to the ADI task. To enhance robustness, we apply additive noise, speed perturbation, frequency masking, and chunk-level dropout. Training is performed with SpeechBrain (Ravanelli et al., 2024) using negative log-likelihood loss, the Adam optimizer, and a NewBob learning rate scheduler starting from 1×10^{-5} for frozen encoder layers and 1×10^{-4} for trainable layers. Training runs for up to 100 epochs on NVIDIA H100 80GB GPUs, with early stopping based on validation performance.

2.3 Experiments and Results

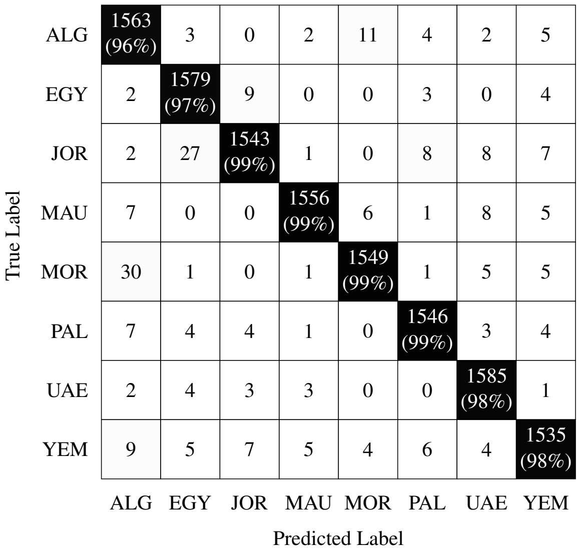
We first evaluated the model after fine-tuning only on ADI-17 and ADI-20-53h to assess zero-shot performance on the NADI validation set. As shown in Table 1, fine-tuning on ADI-17 yields an accuracy of 31.84%, while ADI-20-53h substantially improves zero-shot accuracy to 78.33%.

Fine-tuning dataset	Accuracy (%)
ADI-17	31.84
ADI-20-53h	**78.33**

Table 1: Zero-shot evaluation on the NADI 2025 ADI validation set.

Our final submission builds on the ADI-20-53h model, further adapted with the NADI adaptation split. This two-stage fine-tuning yields substantial gains, as shown in Table 2. The system ranked first, achieving 98.08% accuracy on validation and 79.83% on the test set, with corresponding average costs of 0.0171 and 0.1788 using the 2022 NIST LRE formulation.

Analysis of the validation confusion matrix in 1 shows that the Algerian dialect is the most challenging to predict, with only 96% of utterances correctly classified. Misclassifications primarily involve the geographically adjacent Moroccan dialect, and conversely, 30 Moroccan utterances are labeled as Algerian. Misclassifications between Egyptian and Jordanian are largely reciprocal; despite their geographic proximity, this pattern is unexpected from the perspective of Arabic speakers.



True Label \ Predicted Label	ALG	EGY	JOR	MAU	MOR	PAL	UAE	YEM
ALG	1563 (96%)	3	0	2	11	4	2	5
EGY	2	1579 (97%)	9	0	0	3	0	4
JOR	2	27	1543 (99%)	1	0	8	8	7
MAU	7	0	0	1556 (99%)	6	1	8	5
MOR	30	1	0	1	1549 (99%)	1	5	5
PAL	7	4	4	1	0	1546 (99%)	3	4
UAE	2	4	3	3	0	0	1585 (98%)	1
YEM	9	5	7	5	4	6	4	1535 (98%)

Figure 1: Confusion matrix on the provided development set.

Split	Accuracy (%) ↑	LRE avg. Cost ↓
Validation	98.08	0.0171
Test	79.83	0.1788

Table 2: Final ADI subtask results.

3 Multi-dialectal Arabic ASR

The multi-dialectal Automatic Speech Recognition (ASR) subtask focuses on transcribing spoken Arabic across eight country-level dialects. This subtask aims to highlight the challenges posed by phonetic, lexical, and syntactic diversity of Arabic dialects. In this section, we describe the dataset, model architecture, training methodology, and the obtained results of our approach.

3.1 Dataset

The NADI 2025 ASR dataset includes the same eight dialects as the ADI subtask. Each dialect has 1,600 utterances in both the training and validation splits, with durations ranging from 1 to 30 seconds. The total duration is 30.72 hours (15.44 hours for training, 15.27 for validation). Table 3 shows per-dialect durations.

Dialect	Train (h)	Validation (h)
Algeria	1.91	1.84
Egypt	2.01	1.85
Jordan	1.93	1.89
Mauritania	1.66	1.63
Morocco	1.60	1.67
Palestine	2.43	2.41
UAE	1.87	1.86
Yemen	2.01	2.11
Total	**15.44**	**15.27**

Table 3: Durations per dialect in the NADI 2025 datasets

3.2 Models

We adopted two distinct architectures in our experiments: Whisper and SeamlessM4T-v2 (Barrault et al., 2023). Whisper is an encoder–decoder Transformer model trained on a large-scale multilingual and multitask dataset of speech and text, enabling robust automatic speech recognition (ASR) across a wide range of languages. Its architecture integrates a Transformer-based encoder for speech representation learning and a Transformer decoder for transcription generation. The Whisper-large-v3 model contains approximately 1.55 billion parameters, while Whisper-medium has around 769 million parameters, offering a faster and more memory-efficient alternative.

SeamlessM4T is a multilingual sequence-to-sequence model designed for speech and text translation across more than 100 languages. In its v2 release, it builds upon the UnitY2 architecture, combining a Conformer-based speech encoder with a Transformer-based text decoder. We selected the Egyptian variant due to its demonstrated effectiveness in Arabic transcription tasks. Given the substantial phonetic, lexical, and syntactic divergence between Arabic dialects, we empirically found that fine-tuning a separate model for each dialect outperformed a single unified model for all dialects.

3.3 Experiments and Results

Our experimental process for the multi-dialectal ASR subtask followed three main steps: (i) evaluation of Whisper-based systems, (ii) comparison between per-dialect and unified models, (iii) comparison of the best Whisper model with SeamlessM4T-v2 Large. All results are reported in terms of WER and CER, computed on the NADI 2025 validation and test sets.

For Whisper-based experiments, we fine-tuned different variants (Medium and Large) under two hardware configurations: (i) on an NVIDIA P100 16GB GPU with the AdamW optimizer, a fixed learning rate of 1×10^{-5}, a batch size of 1, and gradient accumulation over 4 steps; and (ii) on an NVIDIA A100 80GB GPU with a batch size of 8 using the same optimizer and learning rate.

For SeamlessM4T, we fine-tuned the v2 Large Egyptian variant for six epochs on an NVIDIA A100 40GB GPU. Training employed the AdamW optimizer with a learning rate warmed up over 100 steps from 1×10^{-9} to 5×10^{-5}. We used label-smoothed negative log-likelihood loss with a smoothing factor of 0.2 and a batch size of 2.

3.3.1 Whisper Large vs Whisper Medium

We first fine-tuned both Whisper-large-v3 and Whisper-medium on the full multi-dialectal dataset (all eight dialects combined). On average, Whisper-large-v3 achieved a WER of 72.20% and a CER of 58.51% while Whisper-medium, despite its smaller size, outperformed it with a WER of 48.21% and a CER of 17.94%. Given these substantial improvements, Whisper-medium was chosen for all subsequent experiments.

3.3.2 Multi vs Mono-dialectal Models

We evaluated two training strategies using Whisper-medium: a multi-dialectal model trained jointly on all dialects, and mono-dialectal models obtained via dedicated fine-tuning for each dialect, yielding eight specialized models. The mono-dialectal approach achieved a lower average Word Error Rate (WER) of 46.71%, compared to 48.21% for the multi-dialectal model. In terms of Character Error Rate (CER), both approaches performed similarly, with the multi-dialectal model scoring 17.97% and the specialized models 17.94% on average. These results suggest that training separate mono-dialectal models is the best approach.

3.3.3 Whisper vs SeamlessM4T-v2 Large

We also compared the best Whisper setup (one specialized whisper-medium system per-dialect) with the SeamlessM4T-v2 Large Egyptian model (Barrault et al., 2023) also fine-tuned separately for each dialect. Due to time constraints, only the large variant was considered for our experiments. Table 4 shows that the Seamless-based system consistently outperforms our best Whisper system for all dialects in both WER and CER.

	Seamless	**Whisper-med.**
Dialect	**WER / CER (%)**	**WER / CER (%)**
Jordan	**25.26 / 7.68**	32.53 / 9.93
Egypt	**30.05 / 12.52**	39.38 / 15.97
Morocco	**39.24 / 13.48**	49.22 / 18.34
Algeria	**54.13 / 19.34**	60.61 / 22.41
Yemen	**50.49 / 16.85**	61.28 / 25.54
Mauritania	**56.93 / 23.91**	62.79 / 26.97
UAE	**30.90 / 10.59**	35.38 / 12.48
Palestine	**26.35 / 9.64**	32.51 / 12.11
Average	**39.17 / 14.25**	46.71 / 17.97

Table 4: SeamlessM4T-v2 Large vs. Whisper-medium WER and CER on the validation sets of each NADI 2025 dialect using one fine-tuned model per dialect.

3.3.4 Official Submission

Based on the validation results presented in table 4, we selected the SeamlessM4T-v2 Large per-dialect models for submission. It ranked second overall, with an average WER 38.54% and CER 14.53% on the test set. Table 5 shows the results per dialect. As it can be seen, performance varied notably across dialects, with Levantine and Egyptian achieving the lowest WERs, while Maghrebi dialects remained the most challenging.

Dialect	**WER (%)**	**CER (%)**
Jordan	28.03	9.36
Egypt	26.83	11.44
Morocco	38.27	13.66
Algeria	53.73	20.43
Yemen	46.63	16.66
Mauritania	58.11	24.53
UAE	29.35	9.91
Palestine	27.36	10.20

Table 5: WER and CER on the NADI 25 test sets.

4 Conclusion

This paper presented the ELYADATA–LIA submissions to the NADI 2025 shared task, addressing both the Arabic Dialect Identification and Multi-dialectal Automatic Speech Recognition subtasks. For ADI, we demonstrated the effectiveness of a two-stage fine-tuning approach using the Whisper-large-v3 encoder, achieving first place with 79.83% accuracy on the test set. For ASR, fine-tuning the SeamlessM4T-v2 Large model separately for each dialect resulted in a strong performance, ranking second on the leaderboard with an average WER of 38.54%.

Acknowledgments

This work was partially funded by the ESPERANTO project. The ESPERANTO project has received funding from the European Union's Horizon 2020 (H2020) research and innovation program under the Marie Skłodowska-Curie grant agreement No 101007666. This work was granted access to the HPC resources of IDRIS under the allocations AD011015051R1 and AD011012551R3 made by GENCI.

References

Bashar Alhafni, Sarah Al-Towaity, Ziyad Fawzy, Fatema Nassar, Fadhl Eryani, Houda Bouamor, and Nizar Habash. 2024. Exploiting dialect identification in automatic dialectal text normalization. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 42–54, Bangkok, Thailand. Association for Computational Linguistics.

Ahmed Ali, Stephan Vogel, and Steve Renals. 2017. Speech recognition challenge in the wild: Arabic mgb-3. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 316–322. IEEE.

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, and 1 others. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1240–1245, Reykjavik, Iceland. European Language Resources Association (ELRA).

Amira Dhouib, Achraf Othman, Oussama El Ghoul, Mohamed Koutheair Khribi, and Aisha Al Sinani. 2022. Arabic automatic speech recognition: A systematic literature review. *Applied Sciences*, 12(17).

Haroun Elleuch, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2025. Adi-20: Arabic dialect identification dataset and models. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, Rotterdam Ahoy Convention Centre, Rotterdam, The Netherlands. To appear.

Charles A. Ferguson. 1959. Diglossia. *WORD*, 15(2):325–340.

Nizar Habash, Mona Diab, and Owen Rambow. 2012. Conventional orthography for dialectal Arabic. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 711–718, Istanbul, Turkey. European Language Resources Association (ELRA).

Nizar Habash, Fadhl Eryani, Salam Khalifa, Owen Rambow, Dana Abdulrahim, Alexander Erdmann, Reem Faraj, Wajdi Zaghouani, Houda Bouamor, Nasser Zalmout, Sara Hassan, Faisal Al-Shargi, Sakhar Alkhereyf, Basma Abdulkareem, Ramy Eskander, Mohammad Salameh, and Hind Saddiki. 2018. Unified guidelines and resources for Arabic dialect orthography. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

James Lane. 2025. The 10 most spoken languages in the world in 2025. https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world. Accessed: 2025-08-12.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, 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, and 14 others. 2024. Open-source conversational AI with SpeechBrain 1.0. *Journal of Machine Learning Research*.

Suwon Shon, Ahmed Ali, Younes Samih, Hamdy Mubarak, and James Glass. 2020. Adi17: A fine-grained arabic dialect identification dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8244–8248. IEEE.

Bashar Talafha, Karima Kadaoui, Samar Mohamed Magdy, Mariem Habiboullah, Chafei Mohamed Chafei, Ahmed Oumar El-Shangiti, Hiba Zayed, Rahaf Alhamouri, Rwaa Assi, Aisha Alraeesi, and 1 others. 2024. Casablanca: Data and models for multidialectal arabic speech recognition. *arXiv preprint arXiv:2410.04527*.

Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarar, Nizar Habash, and Muhammad Abdul-Mageed. 2025. Nadi 2025: The first multidialectal arabic speech processing shared task. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Unicorn at NADI 2025 Subtask 3: GEMM3N-DR: Audio-Text Diacritic Restoration via Fine-tuning Multimodal Arabic LLM

Mohamed Lotfy Elrefai
Ain Shams University
mohamed.lotfy.elrefai@gmail.com

Abstract

We present **GEMM3N-DR**, a multimodal system for NADI 2025 Subtask 3 (*Spoken Arabic Diacritic Restoration*). GEMM3N-DR fine-tunes the *Gemma 3N* LLM via Low-Rank Adaptation (LoRA) using only the official NADI training data, taking both audio and undiacritized text as input and generating fully diacritized output. We apply data augmentation with the nlpaug and the CATT diacritization model. At inference time, we use a structured Arabic instruction and 7-shot examples. Our system achieved a Word Error Rate (WER) of **64%** and Character Error Rate (CER) of **15%** on the hidden test set, ranking **in 2nd place** in the competition. We provide a detailed analysis of model performance, including common error types such as hallucination and incomplete outputs.

1 Introduction

Arabic diacritic restoration is the task of predicting short vowels and other diacritic marks that are omitted in standard Arabic orthography. The problem becomes more challenging in spoken domains, especially for dialectal Arabic, where morphology and phonetics diverge from Modern Standard Arabic (MSA). This task has strong implications for improving readability, ASR post-processing, TTS, and educational tools.

The **NADI 2025 Subtask 3** (Talafha et al., 2025)focuses on diacritic restoration of spoken Arabic dialects using both audio and text. Our approach, **GEMM3N-DR**, leverages the multimodal *Gemma 3N* LLM, adapting it to this task with Low-Rank Adaptation (LoRA) fine-tuning, multi-example prompting, and audio-text fusion.

Our main contributions:

- First application of *Gemma 3N* to spoken Arabic diacritization.
- LoRA-128 fine-tuning with nlpaug-based audio augmentation.
- Use of CATT (Alasmary et al., 2024) predictions as auxiliary inputs for robust training for unlabeled samples, like augment part.
- Structured 7-shot Arabic prompts for inference, reducing WER from 79.05 to 69.05 on devset.

2 Background

The **NADI 2025 Subtask 3: Diacritic Restoration of Spoken Arabic Dialects** (Talafha et al., 2025) challenges participants to restore full Arabic diacritics given an undiacritized transcript and its corresponding speech signal. The task is motivated by the practical need to improve the usability of automatic speech recognition (ASR) outputs, assist language learners, and enhance downstream applications such as text-to-speech (TTS) synthesis.

Task Setup. Participants are given a set of audio-transcript pairs, where transcripts are stripped of diacritics. The goal is to produce fully diacritized text. An example is shown in Table 1.

Input (Undiacritized)	Target (Diacritized)
هذا كتاب جديد	هٰذَا كِتَابٌ جَدِيدٌ

Table 1: Example of task input/output for NADI Subtask 3.

Text-Based Diacritization. Restoring diacritics for written Modern Standard Arabic (MSA) is a well-established problem. Early approaches relied on hand-crafted morphological rules and analyzers, as seen in systems like Madamira (Pasha et al., 2014) and Camelira (Obeid et al., 2022). The field has since evolved through statistical methods to modern deep learning architectures. These include neural sequence-to-sequence models, bidirectional LSTMs followed by Conditional Random

Proceedings of The Third Arabic Natural Language Processing Conference, pages 767–773
November 8-9, 2025 ©2025 Association for Computational Linguistics

Fields (CRF) (Al-Thubaity et al., 2020), and more recently, specialized character-level transformers like **CATT** (Alasmary et al., 2024). A significant recent contribution is **Sadeed** (Aldallal et al., 2025), a decoder-only language model specifically pre-trained and fine-tuned on diverse Arabic corpora. By focusing on high-quality diacritized data, Sadeed demonstrates that specialized models can perform better than general-purpose architectures like CATT, representing a strong benchmark for text-based diacritization.

Audio-Assisted Diacritization. In contrast, the use of audio information to assist in diacritization is a developing field. Text-based models experience a significant performance drop when applied to speech transcripts due to the shift of the domain to the informal spoken language and the prevalence of dialectal variants (Shatnawi et al., 2023). This inadequacy is well documented, with studies showing that speech models trained on gold diacritized data outperform those using text-restored transcripts, highlighting the need for speech-specific solutions (Aldarmaki and Ghannam, 2023).

Pioneering work by (Vergyri and Kirchhoff, 2004) first explored using acoustic information for this task decades ago. Only very recently has this idea been revisited with modern deep learning. Research has branched into complementary approaches: one line of work, exemplified by (Shatnawi et al., 2023), uses a cascaded framework where a fine-tuned Whisper ASR model generates diacritized transcripts to enhance a text-based restoration model. Another approach moves **Beyond Orthography** to directly recover short vowels and dialectal sounds. (Kheir et al., 2024) proposed a novel framework utilizing discrete codes to represent dialectal variability, showing strong performance with limited data and introducing a new dialectal benchmark dataset.

While these methods show promise, they represent disconnected solutions. The former is a cascaded, two-stage pipeline, and the latter focuses on a specific acoustic modeling approach. Our work unifies these directions by proposing a single, end-to-end **multimodal LLM**. Unlike cascaded systems, our model jointly processes raw audio and text signals to directly disambiguate homographs and dialectal variants, effectively bridging the gap between high-quality text diacritization and the challenges of the speech domain.

2.1 Dataset

We used the dataset from the NADI 2025 Shared Task (Subtask 3: Automatic Speech Diacritization) (Talafha et al., 2025), which provides parallel audio-transcript pairs. We participated in the **closed** track is a competition requiring participants to use only the provided resources for a fair comparison. The dataset encompasses a wide range of Arabic varieties and recording conditions, including Dialectal (DIA), Modern Standard (MSA), Classical (CA), and Code-Switched (CS) Arabic.

The training data is composed of two distinct parts:

- **Diacritized Data:** Transcripts with fully vocalized gold standard diacritics (e.g. بَعْدَ الرُّسُومِ الْمُسِيئَةِ لِلنَّبِيِّ ص عَامَ أَلْفَيْنِ وَ سِتَّة تَحَوَّلَتْ حَيَاةُ الْمُسْلِمِينَ فِي الدَّانِمَارْك فَأَصْبَحُوا يَتَمَتَّعُونَ بِكَثِيرٍ مِن لْحُقُوقِ).
- **Non-Diacritized (Augment) Data:** Raw transcripts without diacritics, containing dialectal and code-switched content (e.g., فِين مثلًا أقدر أقول اللي كان يدفعك إنك إنت ما تستسلم هذا التساؤل اللى بينك وبين نفسك إن أنا Senior director.)

2.2 Dataset Statistics

The training set is composed of over 85K sentences drawn from various constituent datasets, each representing a specific Arabic variety. The composition of these datasets is detailed in Table 6. To ensure consistency and quality, samples containing fewer than three words were removed, and punctuation was eliminated from all texts. The resulting dataset consists of 57K samples for training and 1.5K for development (dev), as summarized in Table 2. The training data is further divided into a fully diacritized portion (train) and a partially diacritized portion used for augmentation (augment).

Split	#Utterances	Hours	Avg. Dur. (s)
Train	51517	88.89	6.21
Augment	6087	14.11	8.34
Dev	1580	1.48	3.36
Test	365	0.79	7.83

Table 2: Overall statistics of the NADI 2025 Subtask 3 dataset splits after filtering.

3 System Overview

Our diacritization system is built upon the **Gemma 3N** instruction-tuned language model, which we adapt for the task of Arabic text diacritization using a combination of data augmentation and parameter-efficient fine-tuning. The complete pipeline, from data preparation to final inference, is illustrated in Figure 1 and detailed in the subsequent subsections.

3.1 Augmentation

To enhance the robustness and generalization of our model, we employed a dual-strategy data augmentation approach to effectively increase the size of our training corpus.

- **Audio Augmentation:** Applied a diverse set of audio transformations (pitch shift, noise addition, cropping, speed alteration) using `nlpaug` to enhance acoustic variability, effectively doubling the training data.

- **Text Diacritization:** Utilized the **CATT** model to generate pseudo-labels for non-diacritized text from augmented audio.

3.2 Fine-Tuning

We adapted the pre-trained **Gemma 3N** model to the diacritization task using **LoRA** (Hu et al., 2022).

- **Base Model**: `gemma-3n-E4B-it`

- **PEFT Method**: LoRA

- **Target Modules**: Applied to the key projection matrices within the transformer architecture, specifically targeting both the standard attention mechanisms and audio-specific layers. The targeted modules include:
  - **Attention Projections**: `q_proj`, `k_proj`, `v_proj`, `o_proj`.
  - **Feed-Forward Projections**: `gate_proj`, `up_proj`, `down_proj`.
  - **Audio-Specific Projections**: `post`, `linear_start`, `linear_end`, `embedding_projection`.

- **Hyperparameters**: **Rank** (r): 128, **Alpha** (α): 16, **Dropout**: 0.0

- **Training Setup**: We used the `SFTTrainer` (Supervised Fine-Tuning Trainer).

- **Checkpoint**: The best-performing model was selected from **checkpoint 16500** for final evaluation.

3.3 Inference

At inference time, the model diacritizes raw, non-diacritized Arabic text and audio using a structured prompt-based approach.

- **Prompting**: A fixed **7-shot examples prompt** is used at inference time, consisting of instructions and example pairs.

- **Decoding Parameters**: **Temperature** = 0.001, **Top-*p*** = 1.0 , **Max New Tokens** = 256.

- **Non-Arabic Word Preservation:** Non-Arabic words remain unmodified, maintaining the original sentence structure and ensuring the integrity of code-switched content.

4 Experimental Setup

Our investigation is divided into three primary phases: (1) establishing a baseline performance without any fine-tuning, (2) evaluating parameter-efficient fine-tuning using LoRA, and (3) exploring the effect of increasing the few-shot examples during inference time. All models were evaluated and reported in word error rate (WER% and CER%), where a lower score indicates better performance.

4.1 Baseline Without Fine-tuning

The initial phase establishes a performance baseline for the pre-trained Gemm3n model under two input conditions: using both text and audio data, and using text data alone.

4.2 Fine-tuning With LoRA Parameters

The second phase explores parameter-efficient fine-tuning using LoRA. We experimented with two distinct configurations: a standard LoRA setup with a rank of 8, trained for 5,000 steps, and a more powerful setup combining a high LoRA rank (128) with the 7 few-shot examples identified in the next phase. This aims to quantify the gains from combining advanced fine-tuning techniques with effective prompting.

4.3 Best Fine-tuning Model With Few-Shot Examples At Inference Time

In the Final phase, we investigated the impact of increasing the few-shot examples during inference time on the model (denoted as Gemm3n_F) with a varying number of few-shot examples. The model was evaluated on the development (dev) set, specifically with 3 and 7 examples, to determine if an increased number of few-shot examples improves generalization. The best-performing model checkpoint (at step 16500) was selected for final evaluation to ensure optimal results.

4.3.1 Training Fine-tuning Prompt

We used the following prompt format for training:
System Prompt:
أنت مدقق لغوي لديك ملف صوتي وترى الكلام المكتوب لتخرج أفضل تَشْكِيل للكلمات العربية فقط وأترك الكلام غير العربى كما هو كالإنجليزية والقرنسية على سبيل المثال
User Prompt:
قم بالمراجعة للنص مع المحافظة على نفس عدد الكلمات ولا تخرج كلمات جديدة فقط أضف التشكيلات للكلمات العربية
+ Audio Input
+Text Input without diacritic
Assistant Response:
Label Text without diacritic

4.3.2 Inference Time

We have used similar prompt used in training finetuning with n examples as a few shots.we created a method to determine if the word is non arabic and perserving the position
System Prompt:
أنت مدقق لغوي، لديك ملف صوتي والحروف المكتوبة، أخرج التشكيل الأمثل لكافة الحروف العربية
User Prompt:
رجاءً أضف التشكيل لكل حرف من الحروف العربية في الجملة التالية:

مثال ١:
ذهب محمد إلى المدرسة
ذَهَبَ مُحَمَد إِلَى الْمَدْرَسَةِ .. n examples
النص : Text Input without diacritic
+Audio Input

5 Results and Discussion

The results from our comprehensive experiments are presented below, revealing clear trends regarding the impact of input modalities, few-shot prompting, and parameter-efficient fine-tuning with LoRA.

5.1 Baseline Performance Without Fine-tuning

The initial baseline performance of the pre-trained Gemma model is summarized in Table 3. Contrary to the expectation that multimodal input would enhance performance, the model performed significantly better when processing text-only inputs. The Word Error Rate (WER) for text-only inputs was 71%, a substantial 13 percentage point improvement over the 84% WER achieved with combined text and audio inputs. This result suggests that the pre-trained model may not be effectively leveraging the audio modality; the audio features might be introducing noise or the model's fusion mechanism may be suboptimal for this specific task in a zero-shot setting. It's shown from the table 3 result that the audio representations don't align cleanly with the text task. The model treats irrelevant variations (background noise, accents, prosody) as meaningful, reducing performance. Text-only models are more robust because they avoid this noisy modality.

Model	WER%	CER%	Input Modality
Gemm3n	84	34	Text + Audio
Gemm3n	71	23	Text Only

Table 3: WER and CER on the test set performance with different input modalities without any fine-tuning.

5.2 Impact of LoRA Rank on Performance

Our experiments with LoRA yielded the most significant performance gains, as detailed in Table 4. Applying a standard LoRA configuration (rank=8) for 5,000 steps provided a marginal improvement, reducing the test WER to 82% from the multimodal baseline of 84%. The most effective strategy overall was the combination of a high-capacity LoRA fine-tuning (rank=128) and the 7 few-shot examples are identified in Section 5.3. This configuration achieved a test WER of 64%. This represents a dramatic 20 percentage point improvement over the original multimodal baseline.

Model and LoRA rank	WER%	CER%
Gemm3n_F rank=8	82	35
Gemm3n_F rank=128 + 7-shots	64	15

Table 4: Test set WER and CER after fine-tuning with different LoRA configurations.

5.3 Impact of Few-Shot Examples During Inference

We investigated the impact of providing a varying number of few-shot examples at inference time to the best finetuned model (Gemm3n_F). The results, presented in Table 5, show a clear positive correlation between the number of examples and model performance. Using 7 examples during inference yielded a development set WER of 69.05%, outperforming the configuration with only 3 examples, which achieved a WER of 73.21%. This demonstrates that the model can effectively decrease the hallucination and improve its generalization on the development set.

Model	Few-shots	WER%	CER%
Gemm3n_F	3	73.21	23.22
Gemm3n_F	7	69.05	20.84

Table 5: Deve set WER and CER for the best finetuned model (Gemm3n_F, checkpoint 16500) with a varying number of few-shot examples provided at inference time.

5.4 Analysis of Common Error Types

A qualitative analysis of the predictions from the best-performing model (Gemm3n_F) reveals two primary and distinct error patterns, as illustrated in Table 7 and Table 8.

Table 7 demonstrates the first error type: character-level hallucination and modification. Here, the model does not merely add diacritics but incorrectly alters the base characters themselves (e.g., generating بكَم instead of the reference بكَام). This suggests the model's phoneme-to-grapheme conversion is error-prone, leading to changes in the core lexical items, which is a critical failure mode for a transcription task.

Conversely, Table 8 highlights the second error type: inconsistent diacritization due to data sparsity. For words or syntactic structures likely underrepresented in the training data, the model defaults to a safe, undiacritized output (e.g., أبحث instead of أَبحَثُ). This indicates a failure in generalization and a lack of confidence on unfamiliar patterns.

Our experiments, particularly the improvement from 73.2% to 69.05% WER on the development set by incorporating more diverse few-shot examples, point towards effective strategies to mitigate the observed errors. The performance gain achieved by using examples from different dialects and domains (e.g., formal MSA, Egyptian Arabic, Moroccan Arabic) is significant. This approach directly addresses the error of inconsistent diacritization by providing the model with a richer, more representative context of the task during inference. It acts as a dynamic, in-context learning signal that guides the model towards the desired output style and complexity.

6 Conclusion

In conclusion, our experiments demonstrate that while the pre-trained model struggles with raw multimodal inputs, its performance can be significantly enhanced through a dual approach: (1) parameter-efficient fine-tuning with a high-rank LoRA to adapt the model to the task, and (2) leveraging few-shot examples during inference to provide contextual guidance.

For reproducibility, the implementation and code are available[1] at Unicorn at NADI 2025 Subtask 3.

References

Ahmed Amine Ben Abdallah, Ata Kabboudi, Amir Kanoun, and Salah Zaiem. 2023. Leveraging data collection and unsupervised learning for code-switched tunisian arabic automatic speech recognition. *Preprint*, arXiv:2309.11327.

[1] Full repository URL: https://github.com/MohamedElrefai/GEMM3N-DR-NADI-2025-Subtask-3

Maryam Khalifa Al Ali and Hanan Aldarmaki. 2024. Mixat: A data set of bilingual emirati-English speech. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 222–226, Torino, Italia. ELRA and ICCL.

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*, 8:154984–154996.

Faris Alasmary, Orjuwan Zaafarani, and Ahmad Ghannam. 2024. Catt: Character-based arabic tashkeel transformer. *arXiv preprint arXiv:2407.03236*.

Zeina Aldallal, Sara Chrouf, Khalil Hennara, Mohamed Motaism Hamed, Muhammad Hreden, and Safwan AlModhayan. 2025. Sadeed: Advancing arabic diacritization through small language model. *arXiv preprint arXiv:2504.21635*.

Hanan Aldarmaki and Ahmad Ghannam. 2023. Diacritic recognition performance in arabic asr. *arXiv preprint arXiv:2302.14022*.

Khalid Almeman, Mark Lee, and Ali Abdulrahman Almiman. 2013. Multi dialect arabic speech parallel corpora. In *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6.

Injy Hamed, Ngoc Thang Vu, and Slim Abdennadher. 2020. ArzEn: A speech corpus for code-switched Egyptian Arabic-English. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4237–4246, Marseille, France. European Language Resources Association.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Yassine El Kheir, Hamdy Mubarak, Ahmed Ali, and Shammur Absar Chowdhury. 2024. Beyond orthography: Automatic recovery of short vowels and dialectal sounds in arabic. *arXiv preprint arXiv:2408.02430*.

Ajinkya Kulkarni, Atharva Kulkarni, Sara Abedalmon'em Mohammad Shatnawi, and Hanan Aldarmaki. 2023. Clartts: An open-source classical arabic text-to-speech corpus. In *2023 INTERSPEECH*, pages 5511–5515.

Ossama Obeid, Go Inoue, and Nizar Habash. 2022. Camelira: An arabic multi-dialect morphological disambiguator. *arXiv preprint arXiv:2211.16807*.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T 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 *Lrec*, volume 14, pages 1094–1101.

Sara Shatnawi, Sawsan Alqahtani, and Hanan Aldarmaki. 2023. Automatic restoration of diacritics for speech data sets. *arXiv preprint arXiv:2311.10771*.

Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarar, Nizar Habash, and Muhammad Abdul-Mageed. 2025. Nadi 2025: The first multidialectal arabic speech processing shared task. In *The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou. Association for Computational Linguistics.

Hawau Toyin, Rufael Marew, Humaid Alblooshi, Samar M. Magdy, and Hanan Aldarmaki. 2025. Ar-Voice: A Multi-Speaker Dataset for Arabic Speech Synthesis. In *Interspeech 2025*, pages 4808–4812.

Dimitra Vergyri and Katrin Kirchhoff. 2004. Automatic diacritization of arabic for acoustic modeling in speech recognition.

A Appendix

1.1 Figures

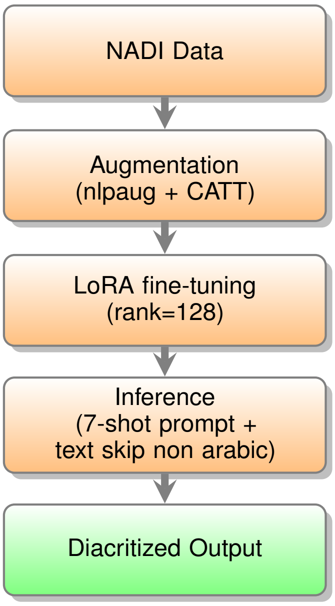


Figure 1: End-to-end pipeline for the diacritization system.

1.2 Tables

Dataset	Type	Diacritized	# Sentences
MDASPC (Almeman et al., 2013)	Multi-dialectal	True	60,677
TunSwitch (Abdallah et al., 2023)	Dialectal, CS	True	5,212
ClArTTS (Kulkarni et al., 2023)	Classical (CA)	True	9,500
ArVoice (Toyin et al., 2025)	MSA	True	2,507
Subtotal		**True**	**77,896**
ArzEn (Hamed et al., 2020)	Dialectal, CS	False	3,344
Mixat (Al Ali and Aldarmaki, 2024)	Dialectal, CS	False	3,721
Subtotal		**False**	**7,065**
Total			**84,961**

Table 6: Breakdown of the constituent datasets within the NADI 2025 original training set.

Reference	Prediction
هُوَّ البُوفِيه المَفتوح بِكَام	هُوَ الْبُوفِيه الْمَفْتُوح بِكَم ؟
هُوَّ فِيه رُسُوم لِلخِدمَه	هُوَ فِيه رُسُومٌ لِلْخِدْمَة

Table 7: Comparison of Model gemm3n_F 7 shots hallucination output compared to Reference by modifying the input text

Reference	Prediction
وَاشُنطُن دِي سِي	واشنطن دي سي
أَنَا أَبحَثُ عَن	أنا أبحث عن

Table 8: Comparison of Model gemm3n_F 7 shots Output against Reference (Undiacritized Samples)

PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic and Islamic Culture

Fakhraddin Alwajih[λ] Abdellah El Mekki[λ] Hamdy Mubarak[γ]
Majd Hawasly[γ] Abubakr Mohamed[γ] Muhammad Abdul-Mageed[λ]
[λ]The University of British Columbia [γ]Qatar Computing Research Institute
{fakhr.alwajih,abdellah.elmekki,muhammad.mageed}@ubc.ca
{hmubarak,mhawasly,abumohamed}@hbku.edu.qa

Subtask 1: General Arabic Culture

This subtask tests large language models on Arabic cultural knowledge. It spans customs, history, arts, and dialects across Arab countries. Models answer multiple-choice questions in Modern Standard Arabic to assess cultural grounding.

Subtask 2: General Islamic Culture

This subtask tests language models on Islamic culture, a core part of Arabic societies. It covers rituals, Quran, Hadith, history, and religious holidays. Models answer multiple-choice questions to assess religious literacy and contextual sensitivity.

Figure 1: An Overview of the PalmX 2025 Shared Task

Abstract

Large Language Models (LLMs) inherently reflect the vast data distributions they encounter during their pre-training phase. As this data is predominantly sourced from the web, there is a high chance it will be skewed towards high-resourced languages and cultures, such as those of the West. Consequently, LLMs often exhibit a diminished understanding of certain communities, a gap that is particularly evident in their knowledge of Arabic and Islamic cultures. This issue becomes even more pronounced with increasingly under-represented topics. To address this critical challenge, we introduce PalmX 2025, the first shared task designed to benchmark the cultural competence of LLMs in these specific domains. The task is composed of two subtasks featuring multiple-choice questions (MCQs) in Modern Standard Arabic (MSA): General Arabic Culture and General Islamic Culture. These subtasks cover a wide range of topics, including traditions, food, history, religious practices, and language expressions from across 22 Arab countries. The initiative drew considerable interest, with 26 teams registering for Subtask 1 and 19 for Subtask 2, culminating in nine and six valid submissions, respectively. Our findings reveal that task-specific fine-tuning substantially boosts performance over baseline models. The top-performing systems achieved an accuracy of 72.15% on cultural questions and 84.22% on Islamic knowledge. Parameter-efficient fine-tuning emerged as the predominant and most effective approach among participants, while the utility of data augmentation was found to be domain-dependent. Ultimately, this benchmark provides a crucial, standardized framework to guide the development of more culturally grounded and competent Arabic LLMs. Results of the shared task demonstrate that general cultural and general religious knowledge remain challenging to LLMs, motivating us to continue to offer the shared task in the future.

1 Introduction

Despite their impressive capabilities, LLMs often display systematic Western- and Anglocentric biases, mirroring the over-representation of these perspectives in their training data (Adilazuarda et al., 2024; Pawar et al., 2025). This lack of cultural diversity can lead to outputs that are not only inappropriate but also harmful. For instance, an Arabic LLM trained on translated English data once suggested having a beer after prayer, a recommen-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 774–789
November 8-9, 2025 ©2025 Association for Computational Linguistics

dation that fundamentally misunderstands (and indeed disrespects) core Arab cultural and religious norms (Naous et al., 2023). Incidents such as this underscore a critical distinction in LLM development between *cultural awareness*, which refers to the understanding of a culture's norms and values and *cultural alignment*, which is focused on the adaptation of actions to respect and reflect these norms and values (AlKhamissi et al., 2024). True progress requires models that are not just culturally aware, but culturally aligned as well.

The need for culturally aligned models is particularly acute in the Arab world, a region of over 450 million people spread across 22 countries. The Arab world comprises immense diversity in customs and traditions, as well as dialectal richness. While recent efforts have produced relatively fluent Arabic LLMs (Bari et al., 2024; Sengupta et al., 2023; Huang et al., 2024), many are trained on machine-translated datasets and evaluated on general NLP tasks in ways that largely overlook country-specific cultural competence. Foundational work on datasets like *Palm* (Alwajih et al., 2025a) has begun to address this by providing culturally inclusive, human-created Arabic instructions covering all 22 Arab countries. However, a standardized benchmark is still needed to systematically measure and compare the cultural understanding of different models.

To bridge this evaluation gap, we introduce the *PalmX 2025* Shared Task, the first benchmarking effort focused specifically on the cultural competence of LLMs in the Arabic context. In this task, we define culture as the collection of knowledge, beliefs, and behaviors encompassing the traditions, social etiquette, cuisine, history, arts, dialectal expressions, and religious practices that characterize communities across the Arab world. *PalmX* challenges models with multiple-choice questions designed to test deep cultural knowledge, not superficial pattern matching. The task is divided into two subtasks: one on *General Arabic Culture* and another on *General Islamic Culture*, reflecting the cornerstones of identity in the region. By providing a standardized evaluation framework, *PalmX* aims to drive the development of LLMs that are not only linguistically fluent but also culturally grounded and respectful.

This paper is organized as follows: Section 2 describes the *PalmX 2025* shared task, including data collection and annotation for both subtasks. Section 3 outlines the participation rules and evaluation methodology. Section 4 presents the participating teams and their results. Section 5 discusses the findings and provides analysis of the methodological approaches for the participating teams. Section 6 concludes with key insights and future directions. Appendix A provides a literature review of related work, and Appendix B presents detailed data analysis including country and topic distributions for datasets of both subtasks.

2 Task Description: PalmX 2025

The objective of the *PalmX 2025* Shared Task[1] is to enable evaluation of the competence of LLMs on Arabic and Islamic cultures through two independent subtasks: *general Arabic culture* and *general Islamic culture*. Each subtask is designed as a set of MCQs in MSA, each with four options (A-D) and a single correct answer; the questions target grounded knowledge. The distractors for each MCQ question are designed to plausible but incorrect, often sharing surface cues to minimize the chance of correct guesses. For each subtask, we provide training, development (dev), and test splits. The training split is provided to participants to support system development, allowing for various approaches such as fine-tuning. Additionally, the dev split is shared with participants to facilitate hyperparameter tuning and local evaluation of their systems before the test phase. The test split is kept private during the competition and is released publicly after the competition concludes. We apply basic quality filters to ensure clarity, a single unambiguous answer, and cultural correctness. This process involves removing off-topic questions unrelated to culture, those with multiple correct answers, biased content, and items with grammatical errors. Accuracy is the primary evaluation metric.

All the resources of *PalmX 2025* shared task are publicly available, including data and evaluation code.[2]

2.1 Subtask 1: General Arabic Culture

The goal of this subtask is to encourage development of methods for incorporating Arabic general culture in LLMs, allowing them to comprehend and reason about diverse aspects of general Arabic culture. These aspects are coming from different cultural categories including *traditional customs*, *local etiquette*, *cuisine*, *historical events*, *famous*

[1] https://palmx.dlnlp.ai/

[2] https://github.com/UBC-NLP/palmx_2025

figures, *geography*, *local languages (dialects)*, and *arts*.

2.1.1 Data Collection and Annotation

The data for this subtask cover a number of cultural topics. To ensure this wide coverage, we follow two complementary data collection strategies, as described below.

Method 1: We source the data from *Palm* (Alwajih et al., 2025a) training split, which we convert into an MCQ format using Qwen3 30B (Yang et al., 2025). Using this method, we acquire $4,000$ samples.

Method 2: We crawl web pages from diverse online resources covering cultural knowledge, customs, etiquette, values, and practices across all Arab countries. Representative sources include *Cultural Crossing*,[3] *Commisceo*[4], *Cultural Atlas*,[5] and *Expatica*.[6] We then segment the collected pages into sections and subsections, and employ GPT-4o-mini to generate culturally relevant MCQs in both Arabic and English. We acquire $1,000$ samples using this method.

For both methods, two professional linguists independently reviewed the data for correctness, removal of low-quality or trivial questions, and acquisition of proper formatting. All discrepancies were reviewed in consolidation sessions. Finally, we shuffle answer options to minimize positional bias.

The final data for this subtask consists of $2,000$, 500, and $2,000$ questions for the training, dev, and test splits, respectively. The domain and country balance in the test set approximates that of the training data but includes some new entities and less frequent cultural items to test generalization.

Samples from Subtask 1 are presented in Table 1.

2.2 Subtask 2: General Islamic Culture

This subtask aims to assess the capacity of LLMs to capture and understand the Islamic culture, which plays a foundational role in Arabic societies. It covers topics such as *Islamic rituals and practices (e.g., prayers and fasting)*, *Quranic knowledge*, *Hadith literature*, *historical developments in Islam*, and *religious holidays*.

2.2.1 Data Collection and Annotation

To enhance topical diversity, we employ two complementary methods to collect Islamic MCQs, yielding a nearly balanced distribution across sources.

Method 1: We create the data based on public Islamic competitions and general questions about Islamic culture using a university book [7]. We acquire 900 samples using this method.

Method 2: We crawl all Islamic articles from *Mawdoo3*, [8] one of the most reputable Arabic content platforms (category: Islam). From this corpus, we randomly select 200 pages and employ GPT-4o-mini to generate diverse MCQs per page. All generated Arabic items are independently reviewed by two professional linguists to verify correctness, eliminate low-quality or trivial content, and ensure proper formatting. Again, all discrepancies are reviewed in consolidation sessions and answer options are subsequently shuffled to reduce positional bias. We acquire $1,000$ samples using this method.

The final data for this subtask consists of 600, 300, and $1,000$ questions for the training, dev, and test splits, respectively.

Samples from Subtask 2 are presented in Table 2.

3 Rules and Evaluation

This section outlines the rules we establish for participation and the methods we employ for the evaluation of submissions. We design the framework to rigorously and fairly assess the intrinsic cultural and Islamic knowledge of the submitted language models.

Reproducibility Teams are instructed to document their data preprocessing, model architecture, external resources, prompt templates, and inference-time strategies.

3.1 Participation and Submission Guidelines

The primary objective of the shared task is to assess the internalized knowledge of LLMs. To ensure the evaluation focuses on the models' core understanding rather than their ability to query external information sources, we established two fundamental rules.

First, the use of systems with real-time data retrieval capabilities, such as retrieval-augmented generation (RAG) or live internet access, is strictly

[3] https://guide.culturecrossing.net/basics_business_student_details.php

[4] https://www.commisceo-global.com/resources/country-guides/

[5] https://culturalatlas.sbs.com.au/countries

[6] https://www.expatica.com/

[7] The Question Bank for Islamic Culture form Al-Balqa Applied University (BAU)

[8] https://mawdoo3.com

Split	Answer	D	C	B	A	Question
train	D	٢٣ سبتمبر	١ يناير	١٤ فبراير	٣٠ نوفمبر	متى يحتفل السعوديون باليوم الوطني؟
train	D	23 September	1 January	14 February	30 November	When do Saudis celebrate National Day?
train	B	الصداقة	الحب	الحزن	الفرح	ماذا ترمز زهور البنفسج في الثقافة الجزائرية؟
train	B	Friendship	Love	Sadness	Joy	What do violets symbolize in Algerian culture?
dev	D	طقس مهم بعد الزفاف	عملية خاصة بالعروس	نوع من الطعام	مباراة تقليدية	ما هو الجرتق في الزواج السوداني؟
dev	D	An important post-wedding ritual	A special process for the bride	A type of food	A traditional contest	What is "Jertiq" in Sudanese weddings?
test	A	الفرنسية	البرتغالية	الإيطالية	الأمازيغية	ما هي اللغة الأم لبعض المغاربة بجانب العربية؟
test	A	French	Portuguese	Italian	Amazigh	What is the mother tongue of some Moroccans besides Arabic?
test	A	المجبوس	الثريد	الكسكس	الكسرة	ما هو الطبق الموريتاني الأكثر شيوعاً في العالم العربي؟
test	A	Majboos	Thareed	Couscous	Kesra	What is the most common Mauritanian dish in the Arab world?

Table 1: Sample questions with their splits, correct answers, and options (A–D) for Subtask 1.

Split	Answer	D	C	B	A	Question
dev	B	رحمة لا تتعلق بالله	رحمة محدودة	رحمة تشمل جميع المخلوقات	رحمة خاصة بالمؤمنين	أي من العبارات التالية تعبر عن معنى اسم الرحمن؟
dev	B	Mercy unrelated to God	Limited mercy	Mercy that includes all creatures	Mercy specific to believers	Which of the following phrases expresses the meaning of the name "Ar-Rahman"?
train	A	عثمان بن عفان	معاذ بن جبل	عبد الرحمن بن عوف	أبو عبيدة بن الجراح	من هو الصحابي الذي لُقّب بأمين هذه الأمة؟
train	A	Uthman ibn Affan	Muadh ibn Jabal	Abdur Rahman ibn Awf	Abu Ubaidah ibn al-Jarrah	Which companion was nicknamed "the trustworthy of this nation"?
test	D	رفيدة بنت سعد الأسلمية رضي الله عنها	حفصة بنت عمر رضي الله عنها	عائشة بنت أبي بكر رضي الله عنها	أم أيمن رضي الله عنها	من هي أول ممرضة في الإسلام؟
test	D	Rufaidah bint Sa'd al-Aslamiyyah (may Allah be pleased with her)	Hafsa bint Umar (may Allah be pleased with her)	Aisha bint Abu Bakr (may Allah be pleased with her)	Umm Ayman (may Allah be pleased with her)	Who was the first nurse in Islam?

Table 2: Sample questions with their splits, correct answers, and options (A–D) for Subtask 2.

prohibited. This ensures that the task does not become a trivial information retrieval challenge. Consequently, submissions are limited to the following format:

1. **Model Weights:** Participants are required to submit the fine-tuned weights of a decoder-only generative language model.

2. **Parameter Limit:** To maintain computational fairness across all participants, the submitted models are constrained to a maximum size of 13 billion (13B) parameters.

3. **Secure Submission:** For privacy and accessibility, participants are instructed to host their models in a private repository on Hugging Face. The final submission consists of the repository ID and a fine-grained access token that provided the organizers with read-only access to the model for evaluation.

Second, to ensure integrity of the results, the test set was held out and remained private to the organizers[9]. This blind evaluation protocol guarantees that no participant had prior access to the test data, enabling a realistic assessment of each model's generalization capabilities in the domain of Arabic cultural and Islamic awareness.

3.2 Evaluation Method

To evaluate the MCQs from our test set, we adopt the likelihood-based method commonly used in frameworks like the EleutherAI Language Model Evaluation Harness (Biderman et al., 2024). This approach assesses a model's understanding by measuring how likely it is to choose the correct answer label after being presented with the question and all possible choices, rather than relying on generative decoding. We develop an in-house script to implement this method and share it with participants during the development phase to ensure they understand how their submissions would be evaluated.

3.2.1 Likelihood-based MCQ Evaluation

For each MCQ item, we construct a prompt that includes the question followed by the list of choices, each prefixed with a letter (e.g., A, B, C, D). The prompt is structured as follows:

[9] Test data was shared only after the leaderboard announcement.

```
<Question>
A. <Choice 1>
B. <Choice 2>
C. <Choice 3>
D. <Choice 4>
Answer:
```

The model's task is to determine which choice label (A, B, C, or D) is the most probable continuation of the prompt. We calculate the likelihood of the model generating each choice label. This approach of scoring only the label, rather than the full text of the choice, ensures the evaluation is not biased by the length of the answer strings.

Specifically, for a given question prompt P and a set of possible choices $\{C_1, C_2, \ldots, C_n\}$, we create n distinct sequences. Each sequence is formed by concatenating the prompt P with the text corresponding to one of the choice labels (e.g., " A", " B", etc.).

Let the tokens for the choice label C_i be $c_{i,1}, c_{i,2}, \ldots, c_{i,k}$. The score for choice C_i is its log-likelihood, calculated as the sum of the conditional log-probabilities of its tokens given the prompt and the preceding tokens of the choice label:

$$\text{score}(C_i) = \log p(C_i|P) = \sum_{j=1}^{k} \log p(c_{i,j}|P, c_{i,1}, \ldots, c_{i,j-1}) \quad (1)$$

These log-likelihood scores are computed for all choices. To select the model's final answer, we normalize these scores into a probability distribution using the softmax function:

$$\text{P}(C_i) = \frac{e^{\text{score}(C_i)}}{\sum_{j=1}^{n} e^{\text{score}(C_j)}}$$

The choice with the highest resulting probability is selected as the model's prediction.

3.2.2 Evaluation Metric

The final performance is measured using **accuracy**. The model's predicted label is compared against the ground-truth label for each question. The overall accuracy is the percentage of questions the model answered correctly:

$$\text{Accuracy} = N_{correct}/N_{total}$$

Where $N_{correct}$ is number of correct predictions and N_{total} is total number of questions. This method provides a robust measure of a model's preference for the correct answer among the given options. The entire process, from prompt construction to likelihood calculation and accuracy scoring, was automated using the provided evaluation script.

4 Shared Task Teams & Results

4.1 Participating Teams

The *PalmX 2025* shared task attracted significant interest from the research community, with 26 teams registering for Subtask 1 (General Culture) and 19 teams registering for Subtask 2 (General Islamic). However, actual participation rates varied between the subtasks. For Subtask 1, eleven teams successfully submitted their models or systems. Among these submissions, two were subsequently rejected due to non-compliance with the established submission guidelines, resulting in nine valid submissions that were evaluated and ranked. For Subtask 2, six teams submitted their approaches, all of which met the submission requirements and were successfully evaluated. Notably, five teams participated in both subtasks, demonstrating their commitment to addressing both domains. This cross-participation allowed for interesting comparisons of team performance across different cultural contexts and question types. Table 3 provides a comprehensive overview of all participating teams, including their subtask involvement and institutional affiliations.

4.2 Baselines

We established baseline performance (accuracy) using the NileChat-3B model (Mekki et al., 2025) without any task-specific fine-tuning (zero-shot):

- **Subtask 1 (General Culture)**: 70.00% on dev and 67.55% on test.
- **Subtask 2 (General Islamic)**: 64.00% on dev and 75.12% on test.

4.3 Shared Task Results

The shared task attracted strong participation, with many teams significantly outperforming the baseline models. This outcome highlights the value of applying task-specific fine-tuning and data augmentation techniques.

Subtask 1: General Arabic Culture

The general culture subtask was exceptionally competitive, with the top four teams finishing within a

Team Name	Affiliation	Subtask 1 (Arabic)	Subtask 2 (Islamic)
HAI (Hossain and Afli, 2025)	ADAPT, MTU	✓	✓
RGIPT (Chatwal and Mishra, 2025)	Rajiv Gandhi Inst. of Petroleum Tech.	✓	
AYA (Tajrin et al., 2025)	Qatar Computing Research Institute	✓	✓
Phoenix (Atou et al., 2025)	Mohammed VI Polytechnic University	✓	✓
CultranAI (Chatwal and Mishra, 2025)	Hamad Bin Khalifa University	✓	
ISL-NLP (Gomaa and Elmadany, 2025)	AAST	✓	
MarsadLab (Biswas et al., 2025)	Hamad Bin Khalifa University	✓	✓
Hamyaria (Al-Dhabyani and Alsayadi, 2025)	Hadhramout Univ., Cairo Univ.	✓	✓
Star (Elrefai et al., 2025)	Alexandria University	✓	
TarnishedLab*	UIR		✓

Table 3: Participating teams, their affiliations, and their subtasks in PalmX 2025. A checkmark (✓) indicates participation in the corresponding subtask. Teams marked with * did not submit their system description papers.

Rank	Name	Accuracy	Model	Size	Dataset(s)	Methodology (concise)
1st	**ADAPT-MTU HAI**	**72.15%**	*NileChat-3B*	**3B**	PalmX (train)	Full fine-tune (CLM); 3 ep; full-prompt supervision.
2nd	**RGIPT**	**71.65%**	*NileChat-3B*	**3B**	PalmX	LoRA (r=16, α=32); 3 ep; no external data.
3rd	**AYA**	**71.45%**	*Fanar-1-9B-Instruct*	**9B**	PalmX Cultural & Islamic (train)	LoRA fine-tune; 3 ep; paraphrase aug (no dev gain).
4th	**Phoenix**	**71.35%**	*Fanar-1-9B-Instruct*	**9B**	PalmX Cultural (train) + LLM aug	FT Fanar-9B with Gemini-based paraphrase/sample/dataset aug (∼18k added).
5th	**CultranAI**	**70.50%**	*Fanar-1-9B-Instruct*	**9B**	PalmX (train+dev), PalmX (test), NativQA MCQs (22k)	LoRA fine-tune; added 22k curated MCQs; train on combined set.
6th	**ISL**	**67.60%**	*NileChat-3B*	**3B**	PalmX Cultural (train)	Retrieval-augmented (Gemini) + PEFT; partial unfreeze of projections.
7th	**MarsadLabM**	**67.55%**	*Qwen2.5-7B-Instruct*	**7B**	PalmX Cultural (train)	LoRA on Qwen2.5-7B (r=16, α=32); 3 ep; 4-bit quantization.
–	**Baseline (ours)**	**67.55%**	*NileChat-3B*	**3B**	–	Zero-shot (no fine-tuning).
8th	**Hamyaria**	**65.90%**	*Qwen2.5-3B-Instruct*	**3B**	PalmX + shuffle/paraphrase aug	Augment (answer shuffle + Fanar-9B paraphrase) + FT Qwen2.5-3B; 5 ep.
9th	**Star**	**64.05%**	*Qwen3-4B*	**4B**	Arabic culture corpus (Wikipedia) + PalmX Cultural	Continual pretrain on Arabic culture corpus; SFT on PalmX with PEFT/LoRA.

Table 4: Approaches for Subtask 1: General Arabic Culture.

narrow 1% accuracy margin.

- **First Place:** The **ADAPT-MTU HAI Team** achieved the top score of **72.15%**. Their strategy involved a full fine-tuning of the NileChat-3B model using a causal language modeling (CLM) objective. They trained the model for three epochs, supervising it over the complete prompt to maximize learning.

- **Second Place:** The **RGIPT Team** secured second place with **71.65%** accuracy. They also used the NileChat-3B model but opted for a parameter-efficient Low-Rank Adaptation (LoRA) approach (r=16, alpha=32). Their model was trained for three epochs on prompt-response pairs derived solely from the provided training data.

- **Third Place:** The **AYA Team** finished third with **71.45%** accuracy. They utilized the larger Fanar-1-9B-Instruct model and experimented with data augmentation by paraphrasing questions with other LLMs. However, this augmentation did not lead to improved performance on the development set, so their final result was based on LoRA fine-tuning for three epochs with a maximum sequence length of 512.

Subtask 2: General Islamic Culture

In the Islamic knowledge subtask, the performance differences between teams were more distinct.

- **First Place:** The **AYA Team** ranked first with a commanding accuracy of **84.22%**, using the ALLaM-7B-Instruct model. Their success stemmed from a combination of effective data augmentation and efficient LoRA fine-tuning, a strategy that proved more successful in the Islamic domain than in the general culture subtask.

- **Second Place:** The **Phoenix Team** took second place with **83.82%** accuracy, also employing the ALLaM-7B-Instruct model. They developed "PhoenixIs" by focusing on paraphrasing for data augmentation and notably included the cultural PalmX dataset in their

Rank	Name	Accuracy	Model	Size	Dataset(s)	Methodology (concise)
1st 🥇	**AYA**	84.22%	*ALLaM-7B-Instruct*	**7B**	PalmX Islamic (train) + aug	LoRA fine-tune on ALLaM-7B with data augmentation.
2nd 🥈	**Phoenix**	**83.82%**	*ALLaM-7B-Instruct*	**7B**	PalmX Islamic (train) + aug + PalmX Cultural	FT ALLaM-7B; paraphrase-focused aug; +Cultural data (∼4.5k).
3rd 🥉	**ADAPT-MTU HAI**	82.52%	*ALLaM-7B-Instruct-preview*	**7B**	PalmX Cultural & Islamic (train)	LoRA (8-bit load); add CoT cue "Let's think step-by-step".
–	**Baseline (ours)**	**75.12%**	*NileChat-3B*	**3B**	–	Zero-shot (no fine-tuning).
4th	**MarsadLabM**	**74.13%**	*Qwen2.5-7B-Instruct*	**7B**	PalmX Cultural	LoRA on Qwen2.5-7B; 3 ep; 4-bit quantization.
5th	**Hamyaria**	**70.83%**	*Qwen2.5-3B-Instruct*	**3B**	PalmX (no aug)	Plain fine-tune on original set; 10 ep.
6th	**TarnishedLab**	**62.84%**	*Qwen2.5-3B-Instruct*	–	–	–

Table 5: Approaches for Subtask 2: General Islamic Culture.

fine-tuning mixture, which expanded their training data to 4,500 questions.

- **Third Place:** The **ADAPT-MTU HAI Team** earned third place with **82.52%** accuracy using the ALLaM-7B-Instruct-preview model. They applied parameter-efficient fine-tuning (LoRA) to an 8-bit loaded version of the model and incorporated reasoning cues like"*Let's think step-by-step*" into their training instances to encourage more structured outputs.

Tables 4 and 5 display the full results for Subtasks 1 and 2, respectively, and briefly describe the system submissions provided by participants, including the backbone models used and their corresponding sizes.

5 Discussion

The results of this shared task provide valuable insights into the current state of Arabic cultural and Islamic knowledge Q&A, revealing several key findings about model performance, methodological approaches, and domain-specific challenges. We discuss a number of these insights here.

5.1 Performance Analysis

The competition demonstrated that task-specific fine-tuning significantly improves performance over baseline models. Most participating teams exceeded the NileChat-3B baseline (67.55% for culture, 75.12% for Islamic), with top performers achieving substantial improvements of 4.6% and 9.1% for Subtasks 1 and 2, respectively. Notably, the Islamic knowledge subtask showed higher overall accuracy scores, with the winning team reaching 84.22% compared to 72.15% for the cultural subtask. This performance difference suggests that Islamic knowledge questions may have more structured, canonical answers compared to the broader host of cultural domains.

5.2 Methodological Insights

Several key methodological trends emerged from the approaches employed by participating teams as we highlight next.

Model selection. Teams favored Arabic-centric models, with NileChat-3B, ALLaM-7B-Instruct, and Fanar-1-9B-Instruct being the most popular choices. Notably, larger models did not necessarily guarantee better performance. This is evidenced by the HAI and RGIPT teams winning the first and second place in subtask 1, respectively, using the smaller NileChat-3B model through effective (parameter-efficient) fine-tuning.

Parameter-efficient fine-tuning. LoRA emerged as the dominant fine-tuning strategy across teams, demonstrating its effectiveness. The success of LoRA-based approaches suggests that efficient adaptation methods can achieve competitive results while maintaining computational feasibility.

Data augmentation strategies. The impact of data augmentation varied significantly between subtasks. While the AYA Team's augmentation approach proved crucial for their success in the Islamic subtask, the same team reported that augmentation did not improve performance on the cultural development set. This suggests that augmentation effectiveness is highly domain- and data-dependent and requires careful study.

Cross-task learning. Teams participating in both subtasks showed varied success patterns. The ADAPT-MTU HAI Team achieved top performance in the cultural subtask but placed third in Islamic questions, while the AYA Team demonstrated the opposite pattern. This indicates that domain expertise and task-specific optimization are crucial factors.

5.3 Domain-Specific Challenges

The performance gap between the two subtasks highlights distinct challenges in Arabic cultural versus Islamic knowledge representation, as follows:

Cultural Knowledge Complexity: The tighter competition in Subtask 1 (top four teams within 1%) suggests that cultural knowledge questions present more nuanced challenges. Cultural information spans diverse topics, regions, and interpretations, making it inherently more complex to model and evaluate.

Islamic Knowledge Structure: The higher accuracies and clearer performance hierarchy in Subtask 2 indicate that Islamic knowledge questions may be slightly less challenging due to being more structured and based on canonical sources and established scholarly consensus. This makes these questions more amenable to current language modeling approaches.

5.4 Technical Innovations

Several technical contributions stood out among the participating teams:

The ADAPT-MTU HAI Team's use of reasoning cues ("Let's think step-by-step") represents an interesting application of chain-of-thought prompting to Arabic cultural domains. The Phoenix team's comprehensive augmentation strategy, exploring paraphrasing, sample-based, and dataset-based approaches, provides valuable insights for future data augmentation research in Arabic NLP.

The ISL-Team's context-aware approach, combining external knowledge retrieval with instruction-based fine-tuning, demonstrates the potential of hybrid architectures for knowledge-intensive tasks in Arabic.

6 Conclusion

The PalmX 2025 Shared Task establishes the first standardized benchmark for evaluating Arabic and Islamic cultural competence in LLMs. Our evaluation framework revealed key insights: task-specific fine-tuning substantially improves performance over baselines, with parameter-efficient approaches (LoRA) emerging as the dominant methodology. The performance gap between cultural (72.15% best) and Islamic knowledge (84.22% best) subtasks suggests domain-specific challenges, with Islamic questions potentially benefiting from more structured canonical sources. Overall, models still struggle on both general cultural and general Islamic knowledge, motivating us to continue to offer the shared task in the future.

Strong community participation from diverse international teams demonstrates the critical need for culturally aligned Arabic LLMs. While participating teams achieved significant improvements over baselines, the modest absolute scores highlight substantial remaining challenges in achieving true cultural competence. PalmX 2025 benchmark provides a foundation for systematic progress tracking and comparison in Arabic cultural AI, driving development of more inclusive language technologies for Arabic-speaking communities worldwide.

Limitations

Several important limitations should be acknowledged:

- **Dataset Imbalances**: PalmX includes data from 22 Arab countries, but the distribution of questions is uneven. Countries like Iraq and Algeria are underrepresented, as shown in the appendix B, while others are overrepresented. This imbalance may bias the models toward frequently represented cultures and limit their generalization to underrepresented communities. Future releases should focus on targeted data collection to improve country-level representation.

- **Evaluation Constraints**: The benchmark is limited to multiple-choice questions in MSA. While this design ensures clarity, fairness, and reproducibility, it does not capture broader aspects of cultural and linguistic competence, such as open-ended reasoning, interactive dialogue, or sensitivity to dialectal variation.

- **Language and Cultural Scope**: PalmX is designed with a focus on Arabic cultural and Islamic knowledge expressed in MSA. However, Arabic speaking communities are linguistically and culturally diverse, with extensive dialectal variation and localized traditions that MSA-based questions may not fully capture. Moreover, Islamic cultural practices extend far beyond the Arab world, but these dimensions are not addressed in a comprehensive way. Therefore, PalmX should be viewed as an initial step toward assessing alignment with Arabic and Islamic cultural contexts, rather than as a complete evaluation of all cultural settings.

- **Quality and Methodology**: Although there were several levels of human review, covering all the dataset used, small sections of the dataset were generated or reformulated using LLMs (see Section 2.1.1), which could introduce subtle artifacts or stylistic biases. Furthermore, the topic classification used for dataset analysis (Appendix B) partially depended on automated methods that have imperfect accuracy. These factors may impact both the reliability of item difficulty and the interpretability of model performance.

Ethical Considerations

The development and evaluation of culturally-aware language models raises several ethical considerations that we have carefully addressed in PalmX 2025:

- **Cultural Representation and Bias**: While we strive for balanced representation across all 22 Arab countries, acknowledged geographical imbalances may inadvertently favor certain cultural perspectives over others. We mitigate this through transparent reporting of data distributions and encourage future work to address underrepresented regions.

- **Religious Sensitivity**: Questions involving Islamic knowledge require particular care to avoid misrepresentation or offense. All religious content was reviewed by qualified experts, and we acknowledge that legitimate scholarly disagreements exist on certain topics. The evaluation framework focuses on widely accepted knowledge rather than contentious interpretations.

- **Data Privacy and Consent**: All data sources used are publicly available or properly licensed. Web-crawled content was limited to public educational resources, and no personal information was collected or used in dataset construction.

- **Model Deployment Implications**: While this benchmark evaluates cultural competence, we emphasize that high performance does not guarantee appropriate real-world deployment. Cultural sensitivity extends beyond factual knowledge to include contextual appropriateness, respect for cultural values, and awareness of power dynamics.

- **Overfitting to Benchmarks**: The competitive nature of shared tasks may unintentionally promote overfitting to benchmark scores rather than fostering genuine cultural competence. As such, it is necessary to stress the importance of engaging with native speakers and experts in addition to the use of benchmarks.

- **Potential Misuse**: A benchmark that evaluates alignment to specific cultural and religious norms could be misapplied in harmful contexts. For instance, it could be used to justify censorship, surveillance, or exclusionary practices. The benchmark data and evaluation methods are designed for research purposes. We encourage responsible use and caution against deploying systems without adequate safeguards for cultural sensitivity and community feedback.

Acknowledgments

Muhammad Abdul-Mageed acknowledges support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[10] and UBC Advanced Research Computing-Sockeye.[11]

References

Muhammad Abdul-Mageed, Abdelrahim Elmadany, Alcides Inciarte, Md Tawkat Islam Khondaker, and 1 others. 2023. Jasmine: Arabic gpt models for few-shot learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16721–16744.

Muhammad Farid Adilazuarda, Sagnik Mukherjee, Pradhyumna Lavania, Siddhant Singh, Alham Fikri Aji, Jacki O'Neill, Ashutosh Modi, and Monojit

[10] https://alliancecan.ca

[11] https://arc.ubc.ca/ubc-arc-sockeye

Choudhury. 2024. Towards measuring and modeling" culture" in llms: A survey. *arXiv preprint arXiv:2403.15412*.

Emran Al-Buraihy, Dan Wang, Tariq Hussain, Razaz Waheeb Attar, Ahmad Ali AlZubi, Khalid Zaman, and Zengkang Gan. 2025. Aratraditions10k bridging cultures with a comprehensive dataset for enhanced cross lingual image annotation retrieval and tagging. *Scientific Reports*, 15(1):19624.

Walid Al-Dhabyani and Hamzah A. Alsayadi. 2025. Hamyaria at PalmX2025: Leveraging Large Language Models to Improve Arabic Multiple-Choice Questions in Cultural and Islamic Domains. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Badr AlKhamissi, Muhammad ElNokrashy, Mai AlKhamissi, and Mona Diab. 2024. Investigating cultural alignment of large language models. *arXiv preprint arXiv:2402.13231*.

Fakhraddin Alwajih, Abdellah El Mekki, Samar Mohamed Magdy, AbdelRahim A. Elmadany, Omer Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, and 1 others. 2025a. Palm: A culturally inclusive and linguistically diverse dataset for Arabic LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32871–32894, Vienna, Austria. Association for Computational Linguistics.

Fakhraddin Alwajih, Samar Mohamed Magdy, Abdellah El Mekki, Omer Nacar, Youssef Nafea, Safaa Taher Abdelfadil, Abdulfattah Mohammed Yahya, Hamzah Luqman, Nada Almarwani, Samah Aloufi, and 1 others. 2025b. Pearl: A multimodal culturally-aware arabic instruction dataset. *arXiv preprint arXiv:2505.21979*.

Fakhraddin Alwajih, El Moatez Billah Nagoudi, Gagan Bhatia, Abdelrahman Mohamed, and Muhammad Abdul-Mageed. 2024. Peacock: A family of Arabic multimodal large language models and benchmarks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12753–12776, Bangkok, Thailand. Association for Computational Linguistics.

Zaid Alyafeai, Khalid Almubarak, Ahmed Ashraf, Deema Alnuhait, Saied Alshahrani, Gubran AQ Abdulrahman, Gamil Ahmed, Qais Gawah, Zead Saleh, Mustafa Ghaleb, and 1 others. 2024. Cidar: Culturally relevant instruction dataset for arabic. *arXiv preprint arXiv:2402.03177*.

Houdaifa Atou, Issam Ait Yahia, and Ismail Berrada. 2025. Phoenix at Palmx: Exploring Data Augmentation for Arabic Cultural Question Answering. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Lama Ayash, Hassan Alhuzali, Ashwag Alasmari, and Sultan Aloufi. 2025. Saudiculture: A benchmark for evaluating large language models' cultural competence within saudi arabia. *Journal of King Saud University Computer and Information Sciences*, 37(6):123.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan Al-Rashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, and 1 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, and 11 others. 2024. Lessons from the trenches on reproducible evaluation of language models. *Preprint*, arXiv:2405.14782.

Md. Rafiul Biswas, Shimaa Ibrahim, Kais Attia, Mabrouka Bessghaier, Firoj Alam, and Wajdi Zaghouani. 2025. MarsadLab at PalmX Shared Task: An LLM Benchmark for Arabic Culture and Islamic Civilization. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Pulkit Chatwal and Santosh Kumar Mishra. 2025. Cultura-Arabica: Probing and Enhancing Arabic Cultural Awareness in Large Language Models via LORA. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Rochelle Choenni and Ekaterina Shutova. 2024. Self-alignment: Improving alignment of cultural values in llms via in-context learning. *arXiv preprint arXiv:2408.16482*.

Eman Elrefai, Esraa Khaled, and Alhassan Ehab. 2025. Star at PalmX 2025: Arabic Cultural Understanding via Targeted Pretraining and Lightweight Fine-tuning. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Sara Ghaboura, Ahmed Heakl, Omkar Thawakar, Ali Husain Salem Abdulla Alharthi, Ines Riahi, Abduljalil Radman, Jorma Laaksonen, Fahad Shahbaz Khan, Salman Khan, and Rao Muhammad Anwer. 2025. CAMEL-bench: A comprehensive Arabic LMM benchmark. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1970–1980, Albuquerque, New Mexico. Association for Computational Linguistics.

Mohamed Gomaa and Noureldin Elmadany. 2025. ISL-NLP at PalmX 2025: Retrieval-Augmented Fine-Tuning for Arabic Cultural Question Answering. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Shehenaz Hossain and Haithem Afli. 2025. ADAPT–MTU HAI at PalmX 2025: Leveraging Full and Parameter-Efficient LLM Fine-Tuning for Arabic Cultural QA. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Song Dingjie, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 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 (Volume 1: Long Papers)*, pages 8139–8163, Mexico City, Mexico. Association for Computational Linguistics.

Rebecca L Johnson, Giada Pistilli, Natalia Menédez-González, Leslye Denisse Dias Duran, Enrico Panai, Julija Kalpokiene, and Donald Jay Bertulfo. 2022. The ghost in the machine has an american accent: value conflict in gpt-3. *arXiv preprint arXiv:2203.07785*.

Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024. Culturellm: Incorporating cultural differences into large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 84799–84838. Curran Associates, Inc.

Zhaoming Liu. 2025. Cultural bias in large language models: A comprehensive analysis and mitigation strategies. *Journal of Transcultural Communication*, 3(2):224–244.

Samar Mohamed Magdy, Sang Yun Kwon, Fakhraddin Alwajih, Safaa Taher Abdelfadil, Shady Shehata, and Muhammad Abdul-Mageed. 2025. JAWAHER: A multidialectal dataset of Arabic proverbs for LLM benchmarking. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12320–12341, Albuquerque, New Mexico. Association for Computational Linguistics.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities. *arXiv preprint arXiv:2505.18383*.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md. Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2025. AraDiCE: Benchmarks for dialectal and cultural capabilities in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4186–4218, Abu Dhabi, UAE. Association for Computational Linguistics.

Tarek Naous, Michael J Ryan, Alan Ritter, and Wei Xu. 2023. Having beer after prayer? measuring cultural bias in large language models. *arXiv preprint arXiv:2305.14456*.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2025. Survey of cultural awareness in language models: Text and beyond. *Computational Linguistics*, pages 1–96.

Abdelrahman Sadallah, Junior Cedric Tonga, Khalid Almubarak, Saeed Almheiri, Farah Atif, Chatrine Qwaider, Karima Kadaoui, Sara Shatnawi, Yaser Alesh, and Fajri Koto. 2025. Commonsense reasoning in Arab culture. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7695–7710, Vienna, Austria. Association for Computational Linguistics.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Jannatul Tajrin, Bir Ballav Roy, and Firoj Alam. 2025. AYA at PalmX 2025: Modeling Cultural and Islamic Knowledge in LLMs. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Yan Tao, Olga Viberg, Ryan S Baker, and René F Kizilcec. 2024. Cultural bias and cultural alignment of large language models. *PNAS Nexus*, 3(9):pgae346.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, and 1 others. 2025. Fanar:

An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Appendices

These appendices provide supplementary material supporting the main findings of this work. The content is organized as follows:

- **A: Literature Review**
  Reviews related work on cultural bias in LLMs, Arabic centric LLMs, and Arabic culturally-Aware datasets and benchmarks.

- **B: Data Analysis**
  This section presents the country-level and topical distributions of both subtasks' datasets.

A Literature Review

Our work is situated at the intersection of several active research areas: the evaluation of cultural biases in LLMs, the development of Arabic-centric models, and the creation of culturally grounded benchmarks.

1.1 Cultural Bias and Alignment in LLMs

The detection, mitigation, and control of cultural bias in LLMs is an expanding research area, seeking to produce generative models that are free of stereotypes and which align with a defined cultural perspective and value framework (Pawar et al., 2025).

Since many LLMs are trained primarily on widely available, high-quality English datasets, they inevitably reflect cultural elements present in those sources (Johnson et al., 2022). Techniques such as fine-tuning and reinforcement learning from human feedback (RLHF) are commonly employed to align such models with a desired value system (Bai et al., 2022; Li et al., 2024); however, this depends on the availability of high-quality instruction data that accurately reflects that system (Liu, 2025). Another approach is to use prompting and system roles to enforce a cultural identity (Tao et al., 2024; Choenni and Shutova, 2024).

1.2 Development of Arabic-Centric LLMs

To counter the dominance of English-centric models, significant efforts have been made to develop foundational LLMs for Arabic. Models like JAIS (Sengupta et al., 2023) pioneered a bilingual Arabic-English training strategy to leverage cross-lingual knowledge transfer. The Jasmine (Abdul-Mageed et al., 2023) suite of models was specifically designed to enhance few-shot learning capabilities in Arabic, while the AceGPT project (Huang et al., 2024) introduced a comprehensive localization pipeline, including pretraining, supervised fine-tuning (SFT), and reinforcement learning with a reward model sensitive to local values.

More recent models like ALLAM (Bari et al., 2024) and Fanar (Team et al., 2025) have further advanced Arabic capabilities. NileChat (Mekki et al., 2025), in particular, was developed as a linguistically diverse and culturally aware model specifically tailored for local communities. NileChat proved that it's possible to build a performant 3 billion parameters language model that can represent the Moroccan and Egyptian communities, including their dialects, cultural heritage, and values through controlled-generated synthetic data. While these models represent crucial advancements in Arabic linguistic competence, their evaluations have largely focused on standard NLP tasks (e.g., question answering, summarization) and general knowledge benchmarks like Arabic MMLU. They have not been systematically evaluated on their understanding of deep, country-specific cultural knowledge.

1.3 Arabic Culturally-Aware Datasets and Benchmarks

A growing body of work is dedicated to developing datasets and benchmarks that reflect Arab culture. One of the earliest benchmark efforts is the Arabic Cultural and Value Alignment dataset (Huang et al., 2024), comprising 8.7K yes–no questions synthetically generated by GPT-3.5 Turbo on various topics related to Arab values. AraDiCE-Culture (Mousi et al., 2025) is a fine-grained benchmark designed to assess cultural awareness across the Gulf, Egypt, and the Levant. Jawaher (Magdy et al., 2025) offers 10K multi-dialectal Arabic proverbs to evaluate understanding of cultural nuances through figurative language. ArabCulture (Sadallah et al., 2025) is a manually crafted dataset of 3.5K commonsense reasoning questions covering the cultures of 13 Arab countries across 54 subtopics.

On the other hand, instruction datasets aimed at embedding cultural understanding during model training include CIDAR (Alyafeai et al., 2024), a 10K culturally localized instruction dataset created

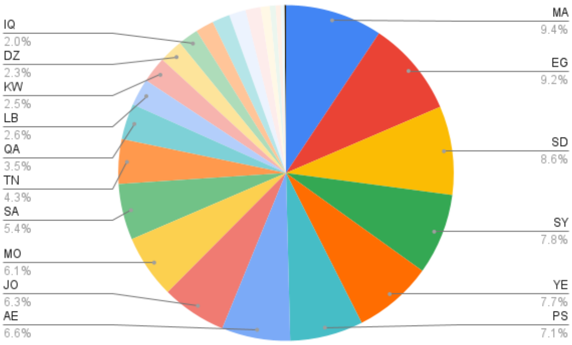


Figure 2: Country distribution of cultural questions in the **training** data.

via machine translation followed by human review, and Palm (Alwajih et al., 2025a), a 17K human-crafted instruction dataset spanning the cultures of the 22 Arab countries. Efforts to support local cultures also include datasets and models such as NileChat (Mekki et al., 2025) for Egyptian and Moroccan dialects, and benchmarks like SaudiCulture (Ayash et al., 2025).

More recently, a focus has emerged on culturally aware Arabic multimodal resources, including Peacock (Alwajih et al., 2024), CamelBench (Ghaboura et al., 2025), AraTraditions10K (Al-Buraihy et al., 2025), and Pearl (Alwajih et al., 2025b).

B Data Analysis

2.1 Subtask 1 Data Analysis

Country distributions of training, development, and test data are shown in Figures 2, 3, and 4. We use ISO 3166 Alpha-2 code for countries[12]. We note that certain countries, such as Iraq (IQ) and Algeria (DZ), are underrepresented across all data splits. In future releases of PalmX, we aim to ensure more balanced country distributions.
Table 6 presents the 15 most frequent topics, which together account for 95% of all test questions, along with illustrative examples. The topics were initially classified using GPT-4o and subsequently consolidated and manually verified. To estimate classification quality, 200 random questions were sampled, yielding an accuracy of 85%. We observe that roughly one-third of the test questions pertain to historical events in Arab countries, such as the dates of revolutions, the founding of political parties, or the birthdates of notable writers.

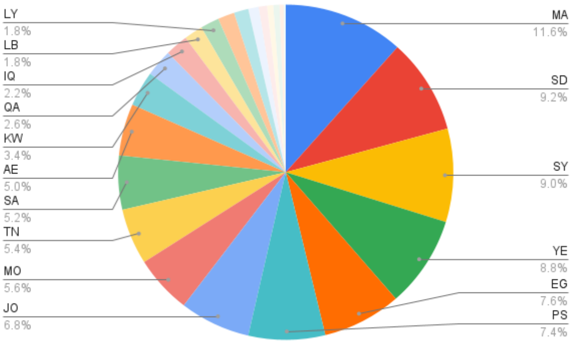


Figure 3: Country distribution of cultural questions in the **development** data.

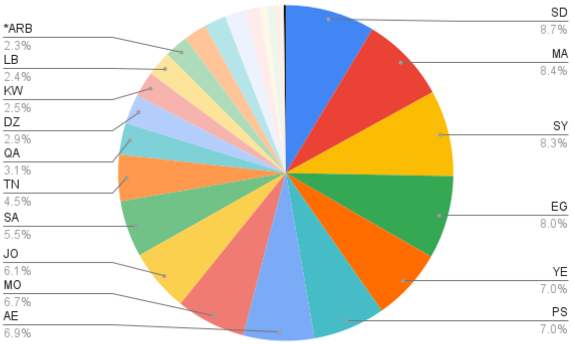


Figure 4: Country distribution of cultural questions in the **test** data. *ARB denotes questions related to Arab culture in general, rather than those tied to a specific country.

[12] https://www.iban.com/country-codes

Topic	Example	%
History	متى تم الاستقلال الجزائري؟ When did Algeria gain independence?	35.2
Geography/Environment	ما هو أكبر الأنهار في سوريا؟ What is the largest river in Syria?	10.0
Food	ما هو طبق البازين في ليبيا؟ What is the Bazin dish in Libya?	7.9
Customs	ما استعمال الحنة في الزواج السوداني؟ What is the use of henna in Sudanese marriage?	7.6
Arts	ما هي أهم فعالية سينمائية في تونس؟ What is the most important cinematic event in Tunisia?	6.0
Sports	ما هي الرياضة الأكثر شعبية في مصر؟ What is the most popular sport in Egypt?	5.1
Literature	متى بدأت الحركة الأدبية الحديثة في قطر؟ When did the modern literary movement begin in Qatar?	4.6
Economics	بماذا تشتهر مدينة بيت لحم في فلسطين من حيث الصناعات؟ What is Bethlehem, Palestine, famous for in terms of industries?	4.6
Religion	ما هو اليوم المقدس للمسلمين في الأسبوع؟ What is the holy day of the week for Muslims?	3.9
Language	ما معنى كلمة «صنطة» في اللهجة العراقية؟ What does the word ’santa’ mean in the Iraqi dialect?	2.8
Clothing	ما هو الطربوش المغربي؟ What is the Moroccan fez?	2.4
Education	ما هي اللغة الثانية التي تُعتبر إلزامية في المدارس الكويتية؟ What second language is mandatory in Kuwaiti schools?	1.5
Politics	من يرأس حزب التجمع من أجل موريتانيا (تمام)؟ Who heads the Rally for Mauritania (RMA) party?	1.3
Tourism	ما يميز شاطئ أرتا في جيبوتي؟ What makes Arta Beach in Djibouti special?	1.3
Law	ما هو السن القانوني للتدخين في البحرين؟ What is the legal smoking age in Bahrain?	1.3
Other 10 topics	Technology, Architecture, Medecine, etc.	5.0

Table 6: Topic distribution of the cultural questions (translated to English) in the **test** set.

2.2 Subtask 2 Data Analysis

Table 7 presents the topic distribution along with examples from the test set. Topic labels were predicted using GPT-4o. To estimate accuracy, we sampled 200 questions and found a 91% agreement with manual annotations. Notably, about one-quarter of the questions concern historical events, such as battles, the birthplaces of scholars, or former names of places.

Topic	Example	%
History	أين وقعت معركة اليرموك؟ Where did the Battle of Yarmouk take place?	25.5
Worship	ما إحدى الفوائد المرتبطة بصلاة الفجر؟ What is one of the virtues of Fajr prayer?	18.2
Ethics	ما أحد مظاهر احترام الآخرين في الإسلام؟ What is one of the manifestations of respecting others in Islam?	12.4
Fiqh (Islamic Jurisprudence)	ما مقدار الزكاة الواجبة في المال؟ How much zakat is due on money?	12.3
Quranic Sciences	ما الآية التي تشير إلى انشقاق القمر؟ Which verse refers to the splitting of the moon?	10.3
Aqidah (Islamic theology)	كم عدد أركان الإيمان؟ How many pillars of faith?	9.4
Hadith Sciences	بماذا يتميز الحديث القدسي؟ What distinguishes the Hadith Qudsi?	3.5
Mu'amalat (Islamic Transactions)	ما الحكم العام للبيع بالتقسيط؟ What is the general ruling on installment sales?	2.4
Contemporary Issues	ما أحد مظاهر التطرف الديني؟ What is one manifestation of religious extremism?	2.1
Sirah (Biography of the Prophet)	من الذي صلى بالناس بعد أن اشتد مرض النبي؟ Who led the people in prayer after the Prophet's illness became severe?	2.0
Philosophy	ماذا يعني مفهوم عالميّة الإسلام؟ What does the concept of the universality of Islam mean?	2.0

Table 7: Topic distribution of the Islamic questions (translated to English) in the **test** set

Hamyaria at PalmX2025: Leveraging Large Language Models to Improve Arabic Multiple-Choice Questions in Cultural and Islamic Domains

Walid Al-Dhabyani
Hadhramout University / Hadhramout, Yemen
Cairo University / Cairo, Egypt
w.aldhabyani@grad.fci-cu.edu.eg

Hamzah A. Alsayadi
Ibb University / Ibb, Yemen
hamzah.sayadi@gmail.com

Abstract

Large language models (LLMs) have been widely used recently. Adapting these models to multiple languages would enhance the accuracy and precision of the other languages. Applying LLMs with Arabic language could improve the prediction of Arabic language. This work applies LLMs with MCQs of Arabic in both the cultural and Islamic domain. The dataset used is PalmX, which is an **MCQ** benchmark dataset. In this work, traditional and AI generation data augmentations are used. For the cultural domain, we applied data augmentation techniques, including paraphrasing using *Fanar-1-9B-Instruct* model and answer shuffling. For the Islamic domain, we used the original dataset without augmentation to maintain content integrity. We then fine-tuned the *Qwen2.5-3B-Instruct* model on both datasets and evaluate its performance, achieving 65.90% accuracy on the cultural set and 70.83% on the Islamic set. Experiment and evaluation are discussed and the best accuracy achieved in this work is explained in both domains.

1 Introduction

Due to their exceptional performance in a wide range of applications, LLMs are becoming more and more well-liked in both academia and industry. Since LLMs are still essential for research and everyday applications, it is becoming more and more important to evaluate them at the task level as well as the societal level in order to better comprehend the hazards they may pose (Chang et al., 2024). Adapting LLMs in Arabic language is still challenging (Mashaabi et al., 2024). Due to grammatical complexity, semantic diversity, and domain specialization, answering multiple choice questions in Arabic is a challenging NLP task, especially in cultural and Islamic contexts. Building strong language-understanding systems requires improving the quality of MCQ datasets in these areas. In order to increase model performance, our work uses LLMs to enhance and optimize Arabic MCQ data. This work was conducted as part of the ArabicNLP 2025 competition named **"PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic Culture"** (Alwajih et al., 2025b)[1].

We use both conventional augmentation methods (answer shuffling) and AI-based methods (paraphrasing using QCRI/Fanar-1-9B-Instruct [2]) (Team et al., 2025) on the general culture dataset. To maintain the authenticity of religion, the Islamic dataset is left unchanged. With significant accuracy gains, Qwen2.5-3B-Instruct [3] (Team, 2024) is refined using both datasets.

The rest of the paper, an introduction is explained in Section 1. Section 2 contains the background. Section 3 illustrated the system overview. Section 4 has the experimental setup. The results are explained in section 5. Finally, conclusion is in section 6.

2 Background

Task setup, dataset details, and related work are explained in this part of the paper.

2.1 Task Setup

Enhancing the quality and precision of Arabic MCQ in two different areas—general culture and Islamic knowledge—is the challenge at hand. Arabic questions with several possible answers make up the input, and choosing the right response from the list of options is the output. Examples about PalmX dataset are in Appendix A.

2.2 Dataset Details

We make use of the PalmX dataset (Alwajih et al., 2025b) which is an MCQ benchmark Arabic dataset. Palmx dataset is created from the Palm

[1] https://palmx.dlnlp.ai/index.html
[2] https://huggingface.co/QCRI/Fanar-1-9B-Instruct
[3] https://huggingface.co/Qwen/Qwen2.5-3B-Instruct

Proceedings of The Third Arabic Natural Language Processing Conference, pages 790–796
November 8-9, 2025 ©2025 Association for Computational Linguistics

Dataset	#Train	#Dev	#Test
Culture	2000	500	2000
Islamic	1000	300	1001

Table 1: PalmX dataset characteristics.

dataset [4] (Alwajih et al., 2025c) (Alwajih et al., 2025a), which is an instruction dataset. The Palm dataset is a comprehensive benchmark created to assess Large Language Models on tasks involving Arabic in a range of dialects and contexts.

The PalmX dataset is especially well-suited to our study goals because it offers broad coverage of both cultural and Islamic knowledge categories. In order to ensure thorough evaluation coverage, the PalmX dataset contains questions covering a range of topics and difficulty levels within each domain. The data set characteristics are illustrated in the table 1. The PalmX dataset has 3000 questions of the training data, 800 questions of the development data and 3001 questions for the test data. The PalmX dataset contains both Cultural and Islamic domains. The separation of both domains are illustrated in table 1.

2.3 Related Work

Previous research in Arabic NLP has highlighted the unique challenges posed by the language's morphological complexity and the need for culturally appropriate content generation. While significant progress has been made in general Arabic NLP tasks, specialized domains such as cultural and Islamic knowledge require targeted approaches that respect content integrity and cultural sensitivities. The remainder of this section is in Appendix B.

The novelty of our contribution lies in the domain-specific approach to Arabic MCQ enhancement, particularly the differentiated treatment of cultural versus Islamic content, recognizing that religious content requires special consideration to maintain authenticity and accuracy.

3 System Overview

In this section, our proposed solutions for both tasks and the used resources are explained in detail. Furthermore, the challenging that we faced through the work with the dataset and other models.

3.1 Key Algorithms and Design Decisions

Our system architecture employs a multi-stage approach involving data preprocessing, augmentation, and model fine-tuning. The core design decision revolves around treating cultural and Islamic domains differently based on their respective requirements for content modification. Figure 1 shows the approaches used in fine-tuning and evaluating the LLMs and Trained Model. In the next two section, we explained the steps used in fine-tunning the modles with Palm dataset.

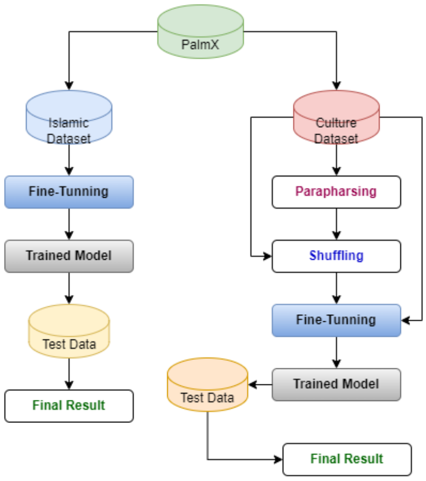


Figure 1: The used model for fine-tunning the LLM for the both dataset (Cultural and Islamic)

3.1.1 Cultural Domain Processing

1. Load original cultural MCQ dataset
2. Generate paraphrased questions using *QCRI/Fanar-1-9B-Instruct*. Paraphrasing techniques follow these approaches:
   - Two questions are generated for each question.
   - Concatenate the original questions with the generated questions separated by the phases " ".
3. Combine original and paraphrased datasets in one dataset.
4. Apply traditional augmentation (answer shuffling). For every question in the new dataset, there are three shuffling in the answers for each question.
5. Apply final answer shuffling, combined the new dataset that contains shuffling answers with the new dataset(orginal dataset + paraphrased dataset)
6. Load the pretrained model *Qwen2.5-3B-Instruct* from Hugging face hub.
7. Fine-tune *Qwen2.5-3B-Instruct* on augmented data.
8. Evaluate accuracy and performance of the

[4] https://github.com/UBC-NLP/palm

Name	Explanation
QCRI/Fanar-1-9B-Instruct	Large language model (LLM) used for paraphrasing the questions.
Qwen2.5-3B-Instruct	Utilized for fine-tuning the model specifically for the Arabic language.
PalmX Dataset	The primary dataset used for model training and evaluation.
Colab A100 GPU (40GB)	Provided the computational resources for experiments and training.
Transformers (Vaswani et al., 2017)	Open-source library used for model loading, fine-tuning, and inference.
PyTorch	Underlying deep learning framework enabling implementation and optimization.

Table 2: Execution environment and resources used in experiments with the PalmX dataset.

trained model in the test dataset.

3.1.2 Islamic Domain Processing

1. Load original Islamic MCQ dataset.
2. Preserve original structure without augmentation.
3. Load the pretrained model *Qwen2.5-3B-Instruct* from Hugging face hub.
4. Fine-tune *Qwen2.5-3B-Instruct* on original Islamic MCQ dataset.
5. Evaluate accuracy and performance of the trained model in the test dataset.

3.2 Resources Used, External Tools and Libraries

The resources used, External Tools and Libraries are explained in table 2. Model access, versioning, and deployment are managed through the *Hugging Face Hub*.

3.3 Addressing Task Challenges

Assuring proper handling of culturally sensitive content, managing the complexity of the Arabic language with its morphological and dialectical variations, striking a balance between the advantages of data augmentation and the preservation of content integrity in domain-specific contexts, and optimizing performance within the limitations of computational resources are the main challenges this work attempts to address. Furthermore, in the paraphrasing stage, there was some words translated to English language that we have addressed and solved.

4 Experimental Setup

In this section of the work, we discussed in detail the experiments steps for both datasets such as Data Split usage, Preprocessing, and Hyperparameter Details, and Evaluation Metrics.

4.1 Data Split Usage

The experiments utilized the standard train/development/test split provided by the PalmX dataset. With evaluation performed directly on the designated test sets for both cultural and Islamic domains.

4.2 Preprocessing and Hyperparameter Details

Cultural Domain Configuration: We fine-tune *Qwen2.5-3B-Instruct* for 1 epoch (optionally extending to `NUM_EPOCHS = 5`) with a learning rate of `1e-5` and a batch size of `1`; for evaluation we use `BATCH_SIZE = 100`. Training uses `8` gradient-accumulation steps, `50` warm-up steps, and a maximum sequence length of `512`, optimized with AdamW (`adamw_torch`) and a cosine learning-rate scheduler. Gradient checkpointing is enabled. Checkpoints are saved every `1,000` steps, evaluation runs every `1,000` steps, and logging occurs every `200` steps.

Islamic Domain Configuration: We likewise use *Qwen2.5-3B-Instruct* for 1 epoch (with extended runs up to `NUM_EPOCHS = 10`) at a learning rate of `1e-5` and a batch size of `1`. The setup includes `8` gradient-accumulation steps, `50` warm-up steps, a maximum sequence length of `512`, the AdamW (`adamw_torch`) optimizer, and a cosine scheduler, with BF16 precision enabled. We save every `300` steps, evaluate every `300` steps, and log every `200` steps.

4.3 Evaluation Metrics

The primary evaluation metric used is accuracy, calculated as the percentage of correctly answered questions in the respective test sets. This metric provides a straightforward measure of model performance in the MCQ answering task. The accuracy, confusion matrix, and heatmap are discussed in detail in this section 5.

5 Results

5.1 Quantitative Findings

Our experiments yielded the following performance results:

Cultural Domain: The model achieved **65.90%** test accuracy using an augmented dataset that combined traditional and AI-based techniques with the

Class	Prec.	Rec.	F1	Sup.
A	0.58	0.77	0.66	497
B	0.67	0.63	0.65	491
C	0.66	0.67	0.66	500
D	0.79	0.57	0.67	512
Acc.			0.66	2000
Macro	0.67	0.66	0.66	2000
W. Avg	0.68	0.66	0.66	2000

Table 3: Classification (confusion matrix) report for the Culture dataset.

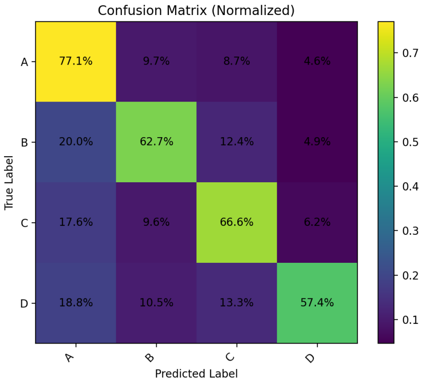


Figure 2: Normalized confusion matrix (heatmap) for cultural dataset

Class	Prec.	Rec.	F1	Sup.
A	0.48	0.73	0.58	153
B	0.89	0.75	0.81	546
C	0.71	0.69	0.70	213
D	0.41	0.56	0.47	73
OTHER	0.00	0.00	0.00	16
Acc.			0.71	1001
Macro	0.50	0.55	0.51	1001
W. Avg	0.74	0.71	0.72	1001

Table 4: Classification report (confusion matrix) for the Islamic dataset.

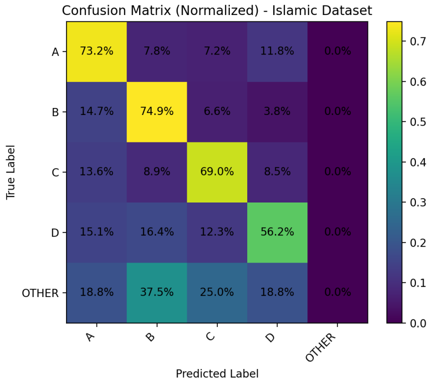


Figure 3: Normalized confusion matrix (heatmap) for Islamic dataset

original data. Training only on the original dataset yielded **65.40%**, while other models performed worse.

Table 3 shows the classification performance. Precision, recall, and F1-scores are balanced across the four classes (A–D) with averages around **0.66**. Class A has high recall (**0.77**) but lower precision (**0.58**), while Class D shows the opposite (**recall 0.57**, **precision 0.79**). The results confirm consistent performance across classes.

The normalized confusion matrix in Figure 2 illustrates the model's classification performance across classes A–D. Correct predictions lie on the diagonal (e.g., **77.1%** of class A and **66.6%** of class C), while off-diagonals show misclassifications (**20%** of class B predicted as A, **18.8%** of class D as A). The model achieves higher recall for classes A and C but struggles with class D, often confused with A (**18.8%**) and C (**13.3%**), suggesting overlapping feature representations between A↔B and D↔A/C.

Islamic Domain: The model achieved **70.83%** test accuracy using the original Islamic dataset without augmentation.

Table 4 summarizes the classification results. Class B performed best (**F1=0.81**, **precision=0.89**, **recall=0.75**), showing robust and balanced performance. Class C also performed well (**F1=0.70**, precision=0.71, recall=0.69). Class A captured many relevant cases with higher recall (**0.73**) but lower precision (**0.48**, F1=**0.58**). Class D underperformed (**F1=0.47**), and the "OTHER" class had no correct predictions due to extremely low support (**16** samples). The overall weighted F1-score is **0.72**, but the lower macro-average F1-score (**0.51**) highlights poorer performance on underrepresented classes.

Figure 3 shows the normalized confusion matrix for the Islamic dataset. Class B achieved the highest recall (**74.9%**), followed by A (**73.2%**) and C (**69.0%**), while Class D had the lowest (**56.2%**). Correct predictions lie on the diagonal, while off-diagonals show key misclassifications: **37.5%** of OTHER samples were predicted as B, **14.7%** of B as A, and **16.4%** of D as B. The "OTHER" class, with very few samples, had no correct predictions, indicating difficulty in recognizing this minority class. Overall, the model performs well on majority classes but struggles with D and OTHER.

5.2 Analysis

The results highlight three key insights: **Domain-Specific Performance:** The Islamic domain achieved higher accuracy (**70.83%**) than the cultural domain (**65.90%**), suggesting that preserving original content structure benefits religious and culturally sensitive queries.

Augmentation Impact: Despite using traditional and AI-based augmentation to expand the cultural dataset, the slightly lower accuracy implies that content preservation can be more critical than dataset size in certain domains. By employing AI-based data augmentation through concatenation of real questions with generated ones, our findings indicate that this approach is not particularly effective. Due to time constraints, we were unable to conduct additional experiments using the original questions combined with the generated ones in a merged dataset, which could potentially improve the accuracy of the trained model. Furthermore, the application of traditional augmentation techniques yielded only marginal benefits. In the context of Arabic MCQ datasets, it is crucial to apply traditional augmentation methods more selectively and precisely. Overall, both augmentation strategies led to an improvement of only 0.5%, which is considered negligible.

Model Configuration: Differences in hyperparameters—particularly more training epochs in the Islamic domain (up to **10** vs. **5**)—may also explain the performance gap.

Model Selection: We experimented with various models for AI-based data augmentation and model training. During the data augmentation phase, we encountered several issues. In particular, many models failed to correctly paraphrase the questions; for example, ALLAM (Bari et al., 2024) often transformed the original questions into different syntactic forms, resulting in outputs that were difficult to interpret. Additionally, some models inadvertently translated certain words into English, even though the questions were primarily in Arabic. For the training phase, we observed that most models produced lower accuracies compared to the QWEN2.5-3B-INSTRUCT model. For instance, LLAMA-3.2-3B-INSTRUCT[5] consistently underperformed relative to QWEN2.5-3B-INSTRUCT.

5.3 Error Analysis

The performance gap between domains suggests several potential factors:

1. **Content Integrity**: Islamic questions may benefit more from maintaining original phrasing and structure due to the precision required in religious knowledge.
2. **Augmentation Effects**: The paraphrasing process in cultural questions might introduce subtle semantic changes that affect answers accuracy.
3. **Training Dynamics**: The different training configurations (more epochs for Islamic domain) may have allowed for better convergence on the Islamic dataset.

5.4 Comparison with Baseline

The Cultural dataset had an accuracy of 65.90%, which was somewhat lower than its baseline of 70% as illustated in (Alwajih et al., 2025a), and the Islamic dataset had an accuracy of 70.83%, which was higher than its baseline of 65%, in comparison to the predetermined baselines. These findings show that the model performed well in the Islamic domain, outperforming the baseline by a significant margin, even while the Cultural dataset performed slightly worse than its baseline. Due to limited computational resources, we were unable to utilize large-scale LLMs such as **Qwen2.5-5B-Instruct** or **Qwen2.5-7B-Instruct**. Nevertheless, we acknowledge that employing such models could potentially yield higher accuracies than those achieved in our experiments.

6 Conclusion

This work presents a comprehensive approach to improving Arabic multiple-choice questions in cultural and Islamic domains using large language models. Our system demonstrates the importance of domain-aware processing, showing that different content domains benefit from tailored approaches to data handling and model training. The key findings indicate that Islamic domain questions achieve better performance when processed without augmentation (70.83% accuracy), while cultural domain questions, despite augmentation efforts, achieve 65.90% accuracy. This suggests that content integrity and cultural sensitivity are paramount considerations when working with specialized Arabic educational content.

Future research directions include the addition of more evaluation techniques. However, we need to investigate transfer learning between Islamic and cultural domains, creating more advanced augmentation techniques that maintain religious and cultural integrity. Extending the method to more effectively handle different Arabic dialects is required. Finally, we need to perform thorough comparisons with other Arabic Models and approaches.

[5] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

References

Abdelrahman Abdallah, Mahmoud Kasem, Mahmoud Abdalla, Mohamed Mahmoud, Mohamed Elkasaby, Yasser Elbendary, and Adam Jatowt. 2024. Arabicaqa: A comprehensive dataset for arabic question answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2049–2059.

Fakhraddin Alwajih, Abdellah El Mekki, Samar Mohamed Magdy, AbdelRahim A. Elmadany, Omer Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, Abdulfattah Mohammed Yahya, Rahaf Alhamouri, Hamzah A. Alsayadi, Hiba Zayed, Sara Shatnawi, Serry Sibaee, Yasir Ech-chammakhy, Walid Al-Dhabyani, Marwa Mohamed Ali, Imen Jarraya, Ahmed Oumar El-Shangiti, Aisha Alraeesi, Mohammed Anwar AL-Ghrawi, Abdulrahman S. Al-Batati, Elgizouli Mohamed, Noha Taha Elgindi, Muhammed Saeed, Houdaifa Atou, Issam Ait Yahia, Abdelhak Bouayad, Mohammed Machrouh, Amal Makouar, Dania Alkawi, Mukhtar Mohamed, Safaa Taher Abdelfadil, Amine Ziad Ounnoughene, Anfel Rouabhia, Rwaa Assi, Ahmed Sorkatti, Mohamedou Cheikh Tourad, Anis Koubaa, Ismail Berrada, Mustafa Jarrar, Shady Shehata, and Muhammad Abdul-Mageed. 2025a. Palm: A culturally inclusive and linguistically diverse dataset for Arabic LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32871–32894, Vienna, Austria. Association for Computational Linguistics.

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025b. PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic and Islamic Culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Fakhraddin Alwajih, Abdellah El Mekki, Samar Mohamed Magdy, Abdelrahim A Elmadany, Omer Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, Abdulfattah Mohammed Yahya, et al. 2025c. Palm: A culturally inclusive and linguistically diverse dataset for arabic llms. *arXiv preprint arXiv:2503.00151*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan Al-Rashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, et al. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Malak Mashaabi, Shahad Al-Khalifa, and Hend Al-Khalifa. 2024. A survey of large language models for arabic language and its dialects. *arXiv preprint arXiv:2410.20238*.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, et al. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A Appendix A

PalmX Dataset Examples

Examples of PalmX dataset for Cultural and Islamic Domains.

Input (Cultural Domain):

- Question: هو ما هي عاصمة مصر؟ (What is the capital of Egypt?)
- Options: أ) القاهرة ب) الإسكندرية ج) الجيزة د) أسوان
- Expected Output: أ) القاهرة

Input (Islamic Domain):

- Question: كم عدد أركان الإسلام؟(How many pillars of Islam are there?)
- Options: أ) ثلاثة ب) أربعة ج) خمسة د) ستة
- Expected Output: ج) خمسة

B Appendix B

Remaining of Related Work

Abdallah et al. (Abdallah et al., 2024) presented "ArabicaQA" which is the first extensive Arabic dataset for open-domain question answering (QA) and machine reading comprehension (MRC). It includes 3,701 difficult unanswerable questions and 89,095 answerable questions, together with open-domain annotations. In addition, the work benchmarks several models, including as GPT-3.5,AraBERT (Antoun et al., 2020), PPLX, and Falcon, on Arabic QA tasks and presents AraDPR, the first dense passage retrieval model trained on Arabic Wikipedia. The results demonstrate that while dense retrieval techniques beat traditional approaches, fine-tuned Arabic-specific models perform better than traditional baselines, but LLMs still have difficulty successfully utilizing retrieved material. Their work advances Arabic natural language processing research by offering empirical insights and a useful resource.

ISL-NLP at PalmX 2025: Retrieval-Augmented Fine-Tuning for Arabic Cultural Question Answering

Mohamed Gomaa, Noureldin Elmadany
Arab Academy for Science, Technology and Maritime Transport
Intelligent Systems Laboratory
Alexandria, Egypt
m.g.abdalla1@student.aast.edu, nourelmadany@aast.edu

Abstract

Cultural understanding is essential for large language models (LLMs), particularly in the Arabic context where many models struggle to capture nuanced cultural elements. To address this gap, we propose a novel approach for Arabic cultural multiple-choice question answering that integrates retrieval-based training data augmentation with parameter-efficient fine-tuning. Our system employs Gemini[1] to retrieve contextual evidence for each question, selects candidate pairs, and adapts NileChat-3B by fine-tuning only three projection layers, reducing trainable parameters by 68.2% while preserving general language proficiency. On the PalmX 2025 Subtask 1 benchmark[2], our system attains 67.60% accuracy on the blind test set, ranking 6th overall and outperforming the NileChat-3B baseline by 3% on the development set. The model weights are publicly available at MohamedGomaa30/Ibn-Al-Nafs.

1 Introduction

The PalmX 2025 (Alwajih et al., 2025) provides a rigorous Arabic cultural benchmark [3]for evaluating AI systems in Arabic, particularly their ability to reason within complex cultural, religious, and historical contexts. This task addresses a key gap in Arabic natural language processing (NLP) by focusing on multiple-choice questions that require cultural reasoning rather than surface-level fact recall.

We tackle this challenge with a two-stage architecture that combines contextual retrieval and parameter-efficient model adaptation, motivated by two observations: (1) Arabic LLMs often lack cultural knowledge available in existing data, and (2) full fine-tuning of large models is computationally expensive and risks catastrophic forgetting of general linguistic abilities.

- **Contextual Retrieval** – We employ Gemini's retrieval features with structured prompts to automatically attach concise (≤50 words) contextual evidence to each question-answer pair in the PalmX 2025 subtask1 dataset. The retrieved evidence captures cultural, geographical, and historical information.
- **Model Adaptation** – We adapt NileChat-3B (Mekki et al., 2025) by fine-tuning only three projection layers: *q_proj* for question representation, *v_proj* for value transformation, and *gate_proj* for information routing. This yields a 68.2% reduction in trainable parameters compared to full fine-tuning.

Our system ranked 6th on the Palmx 2025 leaderboard with 67.70% accuracy on the blind test set, surpassing the NileChat-3B baseline by 3% on the development set. These results demonstrate that targeted architectural choices can improve cultural reasoning in LLMs while preserving computational efficiency and real-world deployability. The remainder of this paper is organized as follows. The background is presented in Section 2. In Section 3, we provided The details of our proposed system are described in Section 3. In Section 4, the experimental results and their analysis are given. Finally, we conclude this paper in Section 5.

2 Background

2.1 Task Setup

The task evaluates the large language model's ability to understand Arabic culture, covering history, geography, arts and customs and traditions for Arabic countries. The input consists of text-based multiple-choice questions (MCQs) and context-aware text that is related to the question that will help the model distinguish the correct answer. This

[1] We use the Gemini 2.5 Pro API for contextual evidence retrieval.
[2] https://example.com/palmx2025
[3] https://palmx.dlnlp.ai

Proceedings of The Third Arabic Natural Language Processing Conference, pages 797–801
November 8-9, 2025 ©2025 Association for Computational Linguistics

is for the training phase only in modern standard Arabic, with the model selecting one correct answer (A, B, C, or D) from four options.

2.1.1 Input Example for Training Phase

ما الملامح الأدبية التي تميز إبداعات محمد الماغوط في السياق
الثقافي السوري؟

Options:

A. تطوير نمط القصة القصيرة الاجتماعية في الصحافة المحلية

B. تأليف روايات تاريخية مستوحاة من الثورة السورية

C. دمج القصيدة النثرية مع المسرح السياسي الساخر في أعماله

D. إحياء الشعر العمودي التقليدي مع إضافة عناصر فلسفية

Context:

يُعرف الماغوط بتأسيسه لـ الشعر الحرْ في سوريا، مبتعدًا عن
عمودية الشعر التقليدي، كما برع في كتابة القصيدة النثرية.
أعماله المسرحية، مثل آشقائق النعمانْ وْكاسك يا وطنْ، اتسمت
بالجرأة والسخرية السياسية، وهي سمة بارزة في الأدب السوري
المعاصر

Output: C

2.1.2 Dataset Preparation

The training dataset is enriched with evidence-based context retrieved through Gemini, which provides historical, geographical, and cultural facts for each multiple-choice pair. This contextual information guides the model in learning cultural cues and improves its ability to select the correct answer.

2.2 Dataset Details

The PalmX 2025 Subtask 1 dataset targets Arabic cultural knowledge, covering customs, traditions, and general background across different Arab countries. The task is evaluated through multiple-choice questions (MCQs), organized as follows:

- **Training Set:** 2,000 MCQ pairs.
- **Development Set:** 500 MCQ pairs for intermediate evaluation.
- **Blind Test Set:** 2,000 unseen MCQ pairs, balanced across countries and domains.

2.3 Related Work

Cultural Alignment in LLMs: Cultural alignment for Large Language Models (LLMs) has received growing attention due to concerns over the dominance of Western perspectives and the marginalization of non-Western cultures (AlKhamissi et al., 2024; Wang et al., 2024). Prior studies show that existing models often fail to capture nuanced cultural variables, leading to irrelevant or biased outputs (Mihalcea et al., 2024; Ryan et al., 2024). This challenge is particularly pronounced for underrepresented linguistic communities such as Arabic speakers, whose cultural diversity is frequently oversimplified (Keleg, 2025).

Arabic Cultural Nuances and LLMs: Several Arabic-centric LLMs have recently been introduced to address these gaps. NileChat-3B is the first Arabic model adapted for Egyptian and Moroccan communities, designed to incorporate dialects, customs, and traditions. Jais (Sengupta et al., 2023) is a bilingual Arabic–English model trained on hundreds of billions of tokens, demonstrating improved reasoning and knowledge in Arabic. AceGPT (Huang et al., 2023) is tailored for Arabic-speaking communities by aligning cultural and linguistic features. Fanar (Abbas et al., 2025) is trained on one trillion Arabic and English tokens and explicitly aligned with Islamic values and Arab cultures. ALLaM (Bari et al., 2024) achieves state-of-the-art performance across several Arabic benchmarks, including Arabic MMLU (Hendrycks et al., 2020), ACVA, and Arabic Exams.

Cultural QA Benchmarks and Technical Adaptation: New benchmarks have advanced cultural evaluation in Arabic NLP, including:

- **ArabicMMLU** (Koto et al., 2024), focusing on educational and academic subjects.
- **ArabDCE-Culture** (Mousi et al., 2024), targeting cultural fact-based QA across diverse Arab countries.
- **BLEnD** (Myung et al., 2024), evaluating everyday Algerian contexts.

From the perspective of model adaptation, improvements in cultural QA have been supported by Parameter-Efficient Fine-Tuning (PEFT) techniques (Xu et al., 2023). Rather than updating all parameters—which is computationally expensive and risks catastrophic forgetting—PEFT updates only a small subset of weights. This reduces memory and compute requirements while enabling targeted adaptation to culturally specific datasets.

3 Proposed System

Our system follows a two-stage pipeline that combines contextual retrieval with parameter-efficient fine-tuning to address the challenges of Arabic cultural multiple-choice question answering (MCQ). In the first stage, we leverage Gemini's. retrieval capabilities to enrich the dataset with culturally relevant evidence. In the second stage, we adapt NileChat-3B through partial fine-tuning of selected layers, reducing computational cost while preserving performance.

3.1 Key Algorithms and Design Decisions

We adopt NileChat-3B as the base model due to its strong performance on Arabic language understanding tasks, particularly in Egyptian and Moroccan contexts. Instead of full fine-tuning—which is computationally expensive and risks catastrophic forgetting—we selectively update only three projection layers: **q_proj** for question representations, **v_proj** for value transformations in attention layers, and **gate_proj** for information routing in feed-forward layers. This strategy reduces trainable parameters by **68.2%** compared to full fine-tuning, improving training efficiency while preserving general linguistic capabilities.

3.2 Resources Beyond Provided Training Data

While PalmX 2025 subtask1 is the primary training dataset, we augmented it with retrieval-augmented context. Using Gemini, we generated concise evidence from trusted cultural, historical, and geographical sources for each MCQ pair. This additional context strengthens the model's ability to make culturally informed decisions beyond surface-level associations.

3.3 Rationale for Training-Time Context Augmentation

The positive effect of training-time context augmentation comes from latent concept alignment rather than memorization. The model is taught to link superficial cues in questions and answers with their underlying cultural principles through the supervisory signal provided by the (question, context, answer) triplets. The internal representations of the model are improved during training in order to encode these patterns of cultural reasoning. As a result, the model exhibits enhanced generalization without explicit context when it is tested, recognizing pertinent cultural cues in unaugmented questions and deducing the right response from its learned conceptual understanding.

3.4 Addressing Task Challenges

The task presents two main challenges. First, Arabic cultural questions require nuanced contextual knowledge beyond factual recall. To address this, we utilized Gemini to retrieve concise, culturally grounded evidence for each MCQ pair, enabling the model to reason with supporting information rather than relying solely on memorization. Second, limited computational resources constrained model training. To mitigate this, we employed parameter-efficient fine-tuning, updating only the **q_proj**, **v_proj**, and **gate_proj** layers of NileChat-3B. This approach reduces computational overhead and mitigates catastrophic forgetting while maintaining strong performance.

3.5 Implementation Details

We implemented our system using **PEFT** [4], the **SFTrainer** from the **TRL** (0.8.2) library [5], and the **Transformers** library (v≥4.41.0) [6]. The dataset was formatted into instruction-response pairs, with a structured Arabic prompt guiding the model to analyze each question, consider candidate answers, and output a single-letter choice (**A**, **B**, **C**, **D**). Training was conducted for three epochs with eight gradient accumulation steps and a per-device batch size of two. To optimize memory and efficiency, we used the **AdamW** optimizer [7], **FP16** mixed precision, a learning rate of 2×10^{-4}, and non-reentrant gradient checkpointing.

4 Experimental Results

4.1 Data Splits

The official `palmx_2025_subtask1_culture` dataset is divided into a training set of 2,000 multiple-choice question–context pairs, a development set of 500 pairs for validation, and a blind test set of 2,000 unseen pairs balanced across cultural domains.

4.2 Data Preprocessing

Each question was augmented with culturally relevant evidence retrieved using Gemini. For every question–answer pair, we constructed a structured prompt in Modern Standard Arabic. The prompt

[4] https://github.com/huggingface/peft

[5] We use the supervised fine-tuning component (SFTrainer) from https://github.com/huggingface/trl

[6] https://github.com/huggingface/transformers

[7] https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html

instructed Gemini to retrieve concise historical, geographical, or cultural evidence (≤50 words) that distinguishes the correct answer from distractors, without explicitly revealing the answer.

4.3 Experimental Settings

We fine-tuned NileChat-3B using the PEFT approach, updating only three projection layers (**q_proj**, **v_proj**, and **gate_proj**). Training was conducted on two NVIDIA T4 GPUs (15 GB each) for three epochs with a per-device batch size of 2, gradient accumulation steps of 8, a learning rate of 2×10^{-4}, and FP16 mixed precision. Optimization used AdamW.

Our implementation relied on the **Transformers** (v≥4.41.0), **TRL** (0.8.2), and **PEFT** libraries from Hugging Face, with dataset handling via **Datasets** and retrieval through Gemini's API. All preprocessing and training scripts will be released publicly for reproducibility.

4.4 Results

We compare our approach on the development split against the model base NileChat-3B to measure the improvement from our method. , and against general-purpose state-of-the-art Arabic models (**Qwen2.5-1.5B** and **Qwen1.5-1.8B**). Using the official metrics of precision, recall, F1-score, and accuracy at Table 1. Our system achieves the best performance across all metrics, with notable improvements over both baselines.The proposed system outperforms Qwen2.5-1.5B by approximately 10% in precision, recall, F1-score, and accuracy, demonstrating its effectiveness.

Model	Pre.	Recall	F1-S	Acc.
Qwen2.5-1.5B	64.73	63.89	63.59	63.60
Qwen1.5-1.8B	63.24	60.88	59.15	59.80
NileChat-3B	71.74	70.00	69.92	70.00
Our system	**73.81**	**73.88**	**73.54**	**73.60**

Table 1: Performance on the development set of PalmX 2025 subtask1, Values are percentages.

5 Conclusion

This paper introduced a parameter-efficient, retrieval-augmented approach for Arabic cultural multiple-choice question answering. Our method combines Gemini-based contextual evidence retrieval with selective fine-tuning of NileChat-3B's projection layers. The approach achieves a **3.0%** improvement over the base model on the development set and ranks **6th** on the official Palmx 2025 leaderboard, showing that targeted architectural adjustments can enhance cultural reasoning while remaining computationally feasible.

However, two limitations remain. First, the cultural knowledge base depends on the coverage and quality of retrieved evidence, which may miss region-specific details. Second, the selective fine-tuning strategy, while efficient, may restrict improvements in tasks requiring broad cross-cultural reasoning or temporal understanding.

Future work will extend the retrieval corpus to cover richer regional variations, integrate temporal reasoning modules for handling historical timelines, and explore hybrid adaptation strategies that combine parameter-efficient fine-tuning with lightweight full-layer updates. These directions aim to further strengthen cultural comprehension in Arabic NLP systems.

5.1 Acknowledgments

We thank the organizers of the Palmx 2025 shared task for providing the dataset and evaluation platform, as well as the anonymous reviewers for their valuable feedback. We also acknowledge the support of the Kaggle platform for providing GPU resources used in this work.

References

Fanar Team Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur A. Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed G. Elfeky, Ahmed K. Elmagarmid, Mohamed Y. Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, Mus'ab Husaini, Soon-Gyo Jung, Ji Lucas, Walid Magdy, Safa Messaoud, Abubakr Mohamed, Tasnim Mohiuddin, Basel Mousi, Hamdy Mubarak, Ahmad Musleh, Zan Ahmad 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*, abs/2501.13944.

Badr AlKhamissi, Muhammad ElNokrashy, Mai Alkhamissi, and Mona Diab. 2024. Investigating cultural alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12404–12422, Bangkok, Thailand. Association for Computational Linguistics.

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and

Muhammad Abdul-Mageed. 2025. PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic and Islamic Culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham Abdullah Alyahya, Sultan Alrashed, Faisal Abdulrahman Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, AbdulMohsen O. Al-Thubaity, Ahmed Abdelali, Jeril Kuriakose, Abdalghani Abujabal, Nora Al-Twairesh, Areeb Alowisheq, and Haidar Khan. 2024. Allam: Large language models for arabic and english. *ArXiv*, abs/2407.15390.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2023. Acegpt, localizing large language models in arabic. In *North American Chapter of the Association for Computational Linguistics*.

Amr Keleg. 2025. LLM alignment for the Arabs: A homogenous culture or diverse ones. In *Proceedings of the 3rd Workshop on Cross-Cultural Considerations in NLP (C3NLP 2025)*, pages 1–9, Albuquerque, New Mexico. Association for Computational Linguistics.

Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. ArabicMMLU: Assessing massive multitask language understanding in Arabic. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5622–5640, Bangkok, Thailand. Association for Computational Linguistics.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities. *ArXiv*, abs/2505.18383.

Rada Mihalcea, Oana Ignat, Longju Bai, Angana Borah, Luis Chiruzzo, Zhijing Jin, Claude Kwizera, Joan Nwatu, Soujanya Poria, and Thamar Solorio. 2024. Why AI is WEIRD and should not be this way: Towards AI for everyone, with everyone, by everyone. *arXiv preprint*, arXiv:2410.16315.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur A. Chowdhury, and Firoj Alam. 2024. Aradice: Benchmarks for dialectal and cultural capabilities in llms. *ArXiv*, abs/2409.11404.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Pérez-Almendros, Abinew Ali Ayele, V'ictor Guti'errez-Basulto, Yazm'in Ib'anez-Garc'ia, Hwaran Lee, Shamsuddeen Hassan Muhammad, Kiwoong Park, Anar Rzayev, Nina White, Seid Muhie Yimam, Mohammad Taher Pilehvar, Nedjma Djouhra Ousidhoum, José Camacho-Collados, and Alice Oh. 2024. Blend: A benchmark for llms on everyday knowledge in diverse cultures and languages. *ArXiv*, abs/2406.09948.

Michael J Ryan, William Held, and Diyi Yang. 2024. Unintended impacts of LLM alignment on global representation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16121–16140, Bangkok, Thailand. Association for Computational Linguistics.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Arun Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Alham Fikri Aji, Zhengzhong Liu, Andy Hock, Andrew Feldman, Jonathan Lee, A. Jackson, Preslav Nakov, Timothy Baldwin, and Eric P. Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *ArXiv*, abs/2308.16149.

Wenxuan Wang, Wenxiang Jiao, Jingyuan Huang, Ruyi Dai, Jen-tse Huang, Zhaopeng Tu, and Michael Lyu. 2024. Not all countries celebrate thanksgiving: On the cultural dominance in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6349–6384, Bangkok, Thailand. Association for Computational Linguistics.

Lingling Xu, Haoran Xie, S. Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *ArXiv*, abs/2312.12148.

ADAPT–MTU HAI at PalmX 2025: Leveraging Full and Parameter-Efficient LLM Fine-Tuning for Arabic Cultural QA

Shehenaz Hossain[1] & Haithem Afli[1]

[1]ADAPT Centre, Computer Science Department,
Munster Technological University, Cork, Ireland

Correspondence: shehenaz.hossain@mymtu.ie, haithem.afli@mtu.ie

Abstract

We present ADAPT–MTU HAI's submission to PalmX 2025, targeting Arabic cultural question answering through large language model (LLM) adaptation. We apply full fine-tuning on NileChat-3B for general cultural comprehension, and parameter-efficient LoRA-based tuning on ALLaM-7B for Islamic knowledge reasoning. Our models achieved first place in the General Culture subtask and third place in the Islamic Culture subtask. This paper outlines our methodology and results, demonstrating the effectiveness of aligning LLM fine-tuning strategies with cultural knowledge domains.

1 Introduction

Language is not merely a tool for communication—it embodies the cultural, historical, and religious identities of its speakers. In Arabic, this interplay is particularly intricate: expressions are shaped by centuries of regional diversity, theological tradition, and social customs (Habash, 2010; Zitouni, 2011; Farghaly and Shaalan, 2009; Darwish et al., 2021). As large language models (LLMs) become increasingly central to NLP applications (Antoun et al., 2020; Touvron et al., 2023; Huang et al., 2024b), a pressing question arises—can these models truly reason over culturally embedded content, especially in linguistically rich and context-dependent settings such as Arabic?

The PaLMX 2025 shared task (Alwajih et al., 2025) [1] directly addresses this challenge through two subtasks. **Subtask 1** focuses on Arabic cultural comprehension, evaluating LLMs on multiple-choice questions (MCQs) covering general cultural knowledge like geography, customs, historical figures, dialectal expressions, and more.

Subtask 2 targets Islamic knowledge reasoning, assessing understanding of Quranic principles, Hadith, and theology. Both subtasks require models to go beyond surface-level fluency and demonstrate genuine cultural and contextual alignment.

Our team submitted systems to both subtasks, building tailored solutions to address their unique requirements. For Subtask 1, we fine-tuned **NileChat-3B** (Mekki et al., 2025), a culturally grounded decoder-only model adapted for North African Arabic under the Language–Heritage–Values (LHV) framework. For Subtask 2, we employed **ALLaM-7B-Instruct** (Bari et al., 2024), an Arabic instruction-tuned model, and applied **parameter-efficient fine-tuning** using LoRA (Brown et al., 2020) with **8-bit quantization** (Dettmers et al., 2023) to reduce memory usage without sacrificing accuracy.

On the official leaderboard, our systems ranked **first** in Subtask 1 with prompt-aligned full fine-tuning for cultural QA, and **third** in Subtask 2, where efficient adaptation highlighted the strength of lightweight tuning in resource-constrained settings.

This paper presents our unified approach to both subtasks. Section 2 summarizes related work, Section 3 details our methodology and training setups, Section 4 discusses results and analysis, and Section 5 concludes with reflections on cultural modeling in Arabic LLMs.

2 Related work

Research on embedding Islamic cultural knowledge into NLP systems is still emerging, though select initiatives have begun to address this need (Saadaoui et al., 2024). The Qur'an QA Shared Task (Malhas et al., 2022, 2023)[2][3] introduced the Qur'anic Reading Comprehension Dataset (QRCD), composed of approximately 1,093 question-passage pairs derived from the Holy Qur'an in Modern Standard Arabic. Participating systems, including AraBERT-based

[1] https://palmx.dlnlp.ai/

[2] https://sites.google.com/view/quran-qa-2022

[3] https://sites.google.com/view/quran-qa-2023

Proceedings of The Third Arabic Natural Language Processing Conference, pages 802–808
November 8-9, 2025 ©2025 Association for Computational Linguistics

models, achieved modest Exact Match (EM) scores below 35%, highlighting the challenge of reasoning over sacred religious text (Mostafa and Mohamed, 2022).

Following this, Hajj-FQA (Aleid and Azmi, 2025) was released in 2025 as the first Arabic dataset targeting pilgrimage-related fatwa questions, offering realistic legal and religious Q&A reflective of common Hajj scenarios (Alyemny et al., 2023). The Hadith-QA corpus expands Islamic QA further by focusing on Prophetic narrations, while IslamicPCQA provides a rich Persian multi-hop benchmark (12,282 QA pairs) over Islamic encyclopedic content, illustrating cross-lingual interest in knowledge reasoning even beyond Arabic contexts (Ghafouri et al., 2023). Recent work has also introduced large-scale QA resources for deep religious understanding, (Qamar et al., 2024) presented a 73,000-question dataset spanning Quranic Tafsir and Ahadith, enriched with contextual explanations and interpretations to support nuanced QA system development.

Additionally, the CAMeL cultural bias benchmark evaluates Arabic LLMs' performance on culturally sensitive prompts, confirming consistent issues with Western-centric bias and cultural misalignment in language models (Naous et al., 2024). In recent years, several Arabic and Arabic-English LLMs have been introduced — including FANAR (Team et al., 2025), JAIS (Sengupta et al., 2023), AceGPT (Huang et al., 2024a), and ALLaM (Bari et al., 2024).In parallel, Arabic cultural and dialectal (Hossain et al., 2025; de Francony et al., 2019) evaluation benchmarks such as CAMELEVAL (Qian et al., 2024) and ARADICE (Mousi et al., 2024) have foregrounded the importance of cultural alignment, dialect robustness, and domain sensitivity in LLM evaluation—factors directly relevant to legal-religious reasoning While these models demonstrate impressive general reasoning and instruction-following ability, independent evaluations reveal that they still inherit cultural biases and struggle with nuanced religious and historical content. For example, (Mohammed et al., 2025) show that even GPT-4 can produce factually incorrect or inconsistent responses to Islamic content due to misinterpreting context, lacking grounding in authoritative sources, and being sensitive to minor wording changes. Similarly, (Alnefaie et al., 2023) report that GPT-4 struggles with Quranic questions, largely because of challenges in classical Arabic, semantic ambiguity, and contextual interpretation.

Despite the advances, structured MCQ-style benchmarks focused specifically on Islamic cultural literacy in Arabic remain rare. PalmX2025 addresses this gap directly, framing cultural understanding explicitly as a multiple-choice reasoning format — making it one of the first shared tasks to assess not just fluency but deep cultural and theological accuracy.

3 Dataset Composition

3.1 Subtask 1: Arabic Cultural Comprehension

This dataset contains culturally grounded MCQs in Modern Standard Arabic on customs, history, geography, arts, cuisine, and dialects, each with four options (A–D) and one correct answer. It includes 2,000 training, 500 development, and 2,000 blind test questions.

3.2 Subtask 2: Islamic Knowledge Reasoning

This dataset contains MCQs on Islamic practices, theology, Quranic knowledge, jurisprudence, and historical context, following the same format as Subtask 1. For training, we combined 600 Subtask 2 MCQs with 2,000 from Subtask 1 to leverage shared linguistic patterns and reasoning structures. It includes 300 development and 1,000 blind test questions.

4 Methodology

4.1 Subtask 1: Full Fine-Tuning of NileChat-3B

For Subtask 1, which focuses on Arabic cultural comprehension, we employ **NileChat-3B** (Mekki et al., 2025)[4], a 3-billion-parameter decoder-only language model built upon Qwen-2.5. NileChat-3B has been instruction-tuned on Egyptian and Moroccan Arabic under the Language–Heritage–Values (LHV) framework, enabling it to capture culturally nuanced responses across Arabic dialects. The model natively supports both Arabic script and Arabizi, making it well-suited for culturally grounded language tasks.

4.1.1 Input Formatting and Tokenization

To ensure strict compatibility with the shared task's evaluation pipeline, each training example is formatted using the official multiple-choice question (MCQ) template provided by the organizers. The

[4] https://huggingface.co/UBC-NLP/NileChat-3B

input consists of a question followed by four answer options prefixed with "A." through "D.", and concludes with the Arabic keyword used to prompt the model's autoregressive completion:

{question text}

A. {option A}
B. {option B}
C. {option C}
D. {option D}

الجواب:

This formatting aligns precisely with the evaluation script, which expects the model to autoregressively generate a single-letter label (e.g., "A") immediately following الجواب:.

Tokenization is performed using the model's associated AutoTokenizer, with inputs truncated or padded to a maximum length of 512 tokens. As the tokenizer does not define a dedicated padding token, we explicitly assign the end-of-sequence token (eos_token) as the pad_token to ensure consistency in attention masking and loss computation across batches.

4.1.2 Training Configuration

Fine-tuning is conducted on a single NVIDIA A100 (40GB) GPU using Hugging Face's Trainer with BF16 precision for 3 epochs, batch size 1, and gradient accumulation of 16 (effective batch size 16). Inputs are truncated or padded to 512 tokens, with full-sequence supervision achieved by copying input_ids into labels and masking padding tokens with -100. This implements standard causal language modeling (CLM), training the model to predict each token from preceding context, including question and answer. We use AdamW (LR 2e–5, no weight decay, without warm-up steps), evaluating and checkpointing at each epoch, and selecting the best model by validation loss. Preprocessing via datasets.map() removes irrelevant columns to reduce memory use and prevent data leakage.

4.2 Subtask 2: LoRA-Based Fine-Tuning of ALLaM-7B

For Subtask 2, which centers on Islamic cultural and legal knowledge reasoning, we adopt **ALLaM-7B-Instruct-preview**(Bari et al., 2024)[5], a 7-billion-parameter Arabic instruction-tuned language model developed to handle Modern Standard Arabic (MSA), Arabic dialects, and culturally grounded textual inputs. Due to its scale and resource requirements, we fine-tune ALLaM-7B using **Low-Rank Adaptation (LoRA)**(Hu et al., 2021), a parameter-efficient approach that significantly reduces memory consumption and training time while preserving task-specific adaptation capabilities.

4.2.1 Input Formatting and Tokenization

To encourage more structured reasoning during training while maintaining compatibility with the evaluation protocol, we introduced an augmented version of this prompt for fine-tuning:

{question text}

A. {option A}
B. {option B}
C. {option C}
D. {option D}

دعنا نفكر خطوة بخطوة:
أجب بالحرف فقط للإجابة الصحيحة (A أو B أو C أو D):
الجواب:

While the evaluation prompt does not contain these (e.g.,:دعنا نفكر خطوة بخطوة) reasoning cues, prior work in prompt engineering has shown that such instructions during fine-tuning can enhance a model's internal reasoning processes without impairing its ability to follow simpler formats at inference(Wei et al., 2022; Kojima et al., 2023). We applied full-sequence causal language modeling (CLM) supervision by duplicating input_ids into labels and used a custom collator for dynamic padding.

4.2.2 LoRA Configuration

To efficiently fine-tune ALLaM-7B, we employ Low-Rank Adaptation (LoRA) using Hugging Face's peft library. Only low-rank matrices injected into the attention projection layers are updated, while the base model remains frozen. Specifically, we target the q_proj and v_proj modules with a LoRA rank of 16, scaling factor (alpha) of 32, and dropout of 0.05.The task is set to CLM, updating under 1% of parameters for efficient adaptation on limited hardware.

4.2.3 Quantization and Memory Optimization

To further reduce GPU memory usage, ALLaM-7B is loaded in 8-bit precision via bitsandbytes and trained in FP16 mixed precision for efficiency. GPU cache clearing and checkpoint pruning control

[5] https://huggingface.co/ALLaM-AI/ALLaM-7B-Instruct-preview

memory usage, with all experiments run on a single NVIDIA RTX 4090 (24GB VRAM).

4.2.4 Training Configuration

Fine-tuning is performed using Hugging Face's Trainer with gradient checkpointing for memory efficiency. Training runs for 5 epochs with a per-device batch size of 8 and gradient accumulation over 4 steps (effective batch size 32), using a maximum sequence length of 256 tokens. Optimization employs AdamW (default β), a learning rate of 3e–5 with cosine decay, 100 warmup steps, and weight decay 0.01. A custom data collator applies dynamic padding and masks padded tokens with -100 to ensure loss is computed only on valid token positions.

4.2.5 Adapter Merging and Deployment

Following fine-tuning, LoRA adapters are merged into the base model resulting in a self-contained checkpoint. The merged model is uploaded to Hugging Face for submission.

4.3 Evaluation Protocol

All final test results were computed by the organizers using the official evaluation script[6] on a held-out blind test set. We submitted our fine-tuned models via Hugging Face, and accuracy was reported based on the organizers' execution of the shared evaluation pipeline.

5 Results

We report results for both subtasks on development and blind test sets (Table 1). Development scores were computed locally with the official evaluation script, while blind test scores were obtained through centralized evaluation by the organizers on a held-out set.

Table 1: Model Accuracy (%) on Development and Test Sets for Both Subtasks

Task	Dev Set (%)	Test Set(Blind) (%)
Subtask 1	78.60	72.15
Subtask 2	75.60	82.52

In Subtask 1, which targets general Arabic cultural awareness, our model achieved 78.60% accuracy on the development set, with a slight drop to 72.15% on the blind test set, likely due to domain shift or question-style variation. For instance, in the development set, it misclassified a question on the main environmental factor affecting the distribution of the Kuhl's free-tailed bat in southwest Saudi Arabia (correct:الحاجة إلى شرب الماء بانتظام) despite predicting heat adaptation, while correctly answering a question on the precise academic trajectory of Dr. Nidal Shamoun in Syria.

Conversely, the Subtask 2 model, which targets domain-specific reasoning in Islamic knowledge, demonstrated strong generalization capacity. Despite a slightly lower dev set performance (75.60%), it achieved a significant improvement on the test set, reaching 82.52%. For example, in one development set question on why a man's testimony equals that of two women, the correct answer was "(B + C) صحيحتان" ("both B and C are correct"); our system selected option B ("النسيان لدى المرأة أكبر من الرجل" – "forgetfulness is greater in women than in men"), which is partially correct but incomplete. In contrast, it correctly answered a question on what is opened for a believer who engages in *tasbīḥ* (تسبيح – "glorification of God"), selecting "أبواب الجنة" ("the gates of Paradise").

These results underscore the methodological rigor of our approach in capturing culturally grounded linguistic patterns under minimal supervision. The coherence between development and test set performance attests to the generalizability and stability of our fine-tuning strategy across evaluation regimes.

6 Conclusion and Future Work

We introduced culturally aligned LLM adaptation strategies that achieved top rankings at PalmX 2025. The combination of full fine-tuning and lightweight LoRA techniques enabled scalable and effective performance across subtasks. In future work, we aim to incorporate retrieval-augmented generation and test robustness on dialectal and low-resource Arabic varieties. Despite these promising results, our approach has limitations. Full fine-tuning is computationally expensive and may not generalise well across domains. Additionally, both datasets are limited in scope, which may affect transferability to unseen topics. Lastly, performance remains sensitive to prompt formatting and initialisation choices, which can impact reproducibility.

Acknowledgments

This research was supported by the ADAPT Research Centre at Munster Technological University.

[6] https://palmx.dlnlp.ai/

ADAPT is funded by Taighde Éireann – Research Ireland through the Research Centres Programme and co-funded under the European Regional Development Fund (ERDF) via Grant 13/RC/2106_P2. We also thank the PalmX 2025 shared task organisers for their efforts in preparing the datasets and evaluation platform, and the anonymous reviewers for their valuable feedback and constructive suggestions, which have helped to improve the quality and clarity of this work.

References

Hani A. Aleid and Aqil M. Azmi. 2025. Hajj-fqa: A benchmark arabic dataset for developing question-answering systems on hajj fatwas. *Journal of King Saud University – Computer and Information Sciences*, 37(6):135.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Is GPT-4 a good islamic expert for answering Quran questions? In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 124–133, Taipei City, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025. PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic and Islamic Culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Ohoud Alyemny, Hend Al-Khalifa, and Abdulrahman Mirza. 2023. A data-driven exploration of a new islamic fatwas dataset for arabic nlp tasks. *Data*, 8(10).

Wissam Antoun, Fady Baly, and Hazem M. Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *CoRR*, abs/2003.00104.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan Al-Rashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, Ahmed Abdelali, Jeril Kuriakose, Abdalghani Abujabal, Nora Al-Twairesh, Areeb Alowisheq, and Haidar Khan. 2024. Allam: Large language models for arabic and english.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab world. *Commun. ACM*, 64(4):72–81.

Gael de Francony, Victor Guichard, Praveen Joshi, Haithem Afli, and Abdessalam Bouchekif. 2019. Hierarchical deep learning for Arabic dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 249–253, Florence, Italy. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing*, 8(4).

Arash Ghafouri, Hasan Naderi, Mohammad Aghajani asl, and Mahdi Firouzmandi. 2023. Islamicpcqa: A dataset for persian multi-hop complex question answering in islamic text resources.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Shehenaz Hossain, Fouad Shammary, Bahaulddin Shammary, and Haithem Afli. 2025. Enhancing dialectal Arabic intent detection through cross-dialect multilingual input augmentation. In *Proceedings of the 4th Workshop on Arabic Corpus Linguistics (WACL-4)*, pages 44–49, Abu Dhabi, UAE. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Song Dingjie, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2024a. 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 (Volume 1: Long Papers)*, pages 8139–8163, Mexico

City, Mexico. Association for Computational Linguistics.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Juncai He, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2024b. Acegpt, localizing large language models in arabic.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

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 *Proceedinsg 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.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities.

Marryam Mohammed, Sama Ali, Salma Khaled, Ayad Majeed, and Ensaf Mohamed. 2025. Aftina: enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.

Ali Mostafa and Omar Mohamed. 2022. GOF at qur'an QA 2022: Towards an efficient question answering for the holy qu'ran in the Arabic language using deep learning-based approach. In *Proceedinsg 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 104–111, Marseille, France. European Language Resources Association.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md. Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2024. Aradice: Benchmarks for dialectal and cultural capabilities in llms.

Tarek Naous, Michael J Ryan, Alan Ritter, and Wei Xu. 2024. Having beer after prayer? measuring cultural bias in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16366–16393, Bangkok, Thailand. Association for Computational Linguistics.

Faiza Qamar, Seemab Latif, and Rabia Latif. 2024. A benchmark dataset with larger context for non-factoid question answering over islamic text.

Zhaozhi Qian, Faroq Altam, Muhammad Alqurishi, and Riad Souissi. 2024. Cameleval: Advancing culturally aligned arabic language models and benchmarks.

Zakia Saadaoui, Ghassen Tlig, and Fethi Jarray. 2024. Llms based approach for quranic question answering. pages 112–118.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models.

Fanar Team, 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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Imed Zitouni. 2011. Introduction to arabic natural language processing n. y. habash. morgan & claypool (synthesis lectures on human language technologies, vol. 10), 2010, xvii+167pp; isbn 978-1-59829-795-9, ebook isbn 978-1-59829-796-6. *Comput. Linguist.*, 37(3):623–625.

A Additional Results

In this appendix, we provide detailed dev set results for both subtasks, comparing baseline (zero-shot) and fine-tuned variants of Fanar-9B-Instruct, NileChat-3B, and ALLaM-7B models. These results illustrate the consistent improvements achieved through fine-tuning, with larger models generally benefiting more from adaptation.

Table 2: Dev set accuracy of baseline and fine-tuned models for subtask 1.

Model	**Fine-tune**	**Dev Acc.(%)**
Fanar-1-9B-Instruct	(zero-shot)	69.80
Fanar-1-9B-Instruct	(fine-tuned)	75.40
NileChat-3B	(zero-shot)	70.00
NileChat-3B	(fine-tuned)	78.60

For Subtask 1 (cultural QA), Fanar-1-9B-Instruct improves from 69.80% in zero-shot to 75.40% after fine-tuning, while NileChat-3B achieves the highest dev accuracy of 78.60% after fine-tuning.

Table 3: Dev set accuracy of baseline and fine-tuned models for subtask 2.

Model	**Fine-tuning**	**Dev Acc.(%)**
NileChat-3B	(fine-tuned)	71.67
ALLaM-7B	(zero-shot)	68
ALLaM-7B	(PEFT)	75.60

For Subtask 2 (Islamic knowledge reasoning), NileChat-3B with fine-tuning reaches 71.67%, while ALLaM-7B shows stronger performance, improving from 68.00% zero-shot to 75.60% after PEFT-based adaptation.

CultranAI at PalmX 2025: Data Augmentation for Cultural Knowledge Representation

Hunzalah Hassan Bhatti[1*], **Youssef Ahmed**[1*], **Md Arid Hasan**[2], **Firoj Alam**[3]
[1]Qatar University, [2]University of Toronto, Canada
[3]Qatar Computing Research Institute
hunzalahhassan@gmail.com, fialam@hbku.edu.qa

Abstract

In this paper, we report our participation to the PalmX cultural evaluation shared task. Our system, ***CultranAI***, focused on data augmentation and LoRA fine-tuning of large language models (LLMs) for Arabic cultural knowledge representation. We benchmarked several LLMs to identify the best-performing model for the task. In addition to utilizing the PalmX dataset, we augmented it by incorporating the Palm dataset and curated a new dataset of over 22K culturally grounded multiple-choice questions (MCQs). Our experiments showed that the Fanar-1-9B-Instruct model achieved the highest performance. We fine-tuned this model on the combined augmented dataset of 22K+ MCQs. On the blind test set, our submitted system ranked 5th with an accuracy of 70.50%, while on the PalmX development set, it achieved an accuracy of 84.1%. We made experimental scripts publicly available for the community.[1]

1 Introduction

Cultural information plays a pivotal role in shaping human identity, behavior, and social interactions. It encompasses the shared beliefs, values, customs, languages, traditions, and collective knowledge of a community or society. In today's interconnected information, communication, and interaction ecosystem, hundreds of millions of users engage with LLMs for everyday queries - many of which involve aspects of local culture, traditions, cuisine, and more (Pawar et al., 2025; Hasan et al., 2025). A central challenge lies in evaluating how effectively LLMs comprehend and generate responses to such culturally embedded queries, particularly in multilingual settings characterized by significant dialectal variation. Other challenges include how to develop culturally aligned LLMs (Wang et al., 2023) and make them available in low-compute environments (Hu et al., 2022). Recent initiatives have introduced evaluation resources - such as culturally relevant datasets, task-specific benchmarks, and performance metrics - to assess LLM capabilities in this domain (Myung et al., 2024; Li et al., 2024b; Mousi et al., 2025).

Yet these efforts remain limited, especially in achieving deeper, dialect-specific advancements. Addressing this gap requires sustained, targeted, rigorous initiatives. The PalmX Shared Task at ArabicNLP 2025 (Alwajih et al., 2025b) is a step in this direction, offering a dedicated benchmark for culturally specific evaluation with a special emphasis on Arabic - thereby advancing the development of LLMs that are both linguistically and culturally aligned. Other recent relevant efforts for Arabic include the development of Arabic-centric LLMs (Team et al., 2025; Sengupta et al., 2023; Bari et al., 2025), leaderboards (Al-Matham et al., 2025), and culturally specific datasets (Alwajih et al., 2025a; Ayash et al., 2025).

To advance the state of the art in Arabic cultural knowledge representation within LLMs, in this paper, we report our participation in the shared task. We specifically focus on the cultural evaluation subtask. To address the challenges of training and deploying LLMs in low-compute resource settings, we conducted a comparative analysis of quantized vs. full-precision models. In parallel, we employed LLM-driven data augmentation strategies to improve the model accuracy. To summarise, the contributions of our study are as follows.

- We provide a performance comparison of different LLMs (Arabic-centric and multilingual) in a zero-shot setup.
- We demonstrate that the performance gap between quantized models and their full-precision counterparts is minimal.
- We show that data augmentation contributes to improving model performance.

[*] The contribution was made while the author was interning at the Qatar Computing Research Institute.

[1] https://github.com/hunzed/CultranAI

Proceedings of The Third Arabic Natural Language Processing Conference, pages 809–817
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Related Work

General Capabilities of LLMs. LLMs have demonstrated remarkable capabilities across a wide range of natural language processing (NLP) tasks, including text classification, question answering, summarization, and dialogue generation (Bubeck et al., 2023; Abdelali et al., 2024). Their ability to leverage vast amounts of pretraining data and adapt to downstream tasks with minimal supervision has enabled strong performance in both zero-shot and few-shot settings (Abdelali et al., 2024). These advances have accelerated the integration of LLMs into diverse real-world applications spanning education, healthcare, finance, and customer support.

Cultural and Everyday Knowledge. Despite successes in several downstream NLP tasks, LLMs often underperform on tasks requiring culturally grounded knowledge, particularly in low-resource languages and dialects (Pawar et al., 2025; Hasan et al., 2025; Alam et al., 2025). A culturally aligned model should accurately interpret and generate content that reflects local linguistic forms, social norms, and lived experiences across domains such as healthcare, education, and cuisine (Li et al., 2024b,a; Shi et al., 2024). However, current models frequently fail to capture region-specific expressions and indigenous knowledge, limiting their effectiveness in culturally nuanced contexts (Myung et al., 2024; Chiu et al., 2025). To address these limitations, recent research has focused on developing benchmarks and datasets that evaluate and enhance LLMs' performance for both cultural and everyday information-seeking queries. These resources span mono- and multilingual settings and are sourced from diverse origins, including Wikipedia (Yang et al., 2018; Kwiatkowski et al., 2019), Google Search QA (Khashabi et al., 2021), Reddit forums (Fan et al., 2019), and native speaker-authored question–answer pairs (Clark et al., 2020). Other approaches combine native and machine-translated content or employ LLMs to generate culturally relevant QA datasets (Putri et al., 2024; Li et al., 2024b).

Although English and multilingual resources have advanced the state of the art in culturally aligned LLMs, the richness and diversity of the Arabic language and its dialects require dedicated efforts in both resource creation and culturally aligned model development. Recent initiatives have begun addressing this gap through the development of datasets for benchmarking and fine-tuning Arabic-centric models (Mousi et al., 2025; Alwajih et al., 2025a). The PalmX Shared Task at ArabicNLP 2025 is a targeted initiative to advance culturally aligned LLM development through a benchmark for culturally grounded evaluation in Arabic.

3 Task and Dataset

3.1 Task Overview

The PalmX 2025 shared task offered two subtaks, one of which is *General Culture Evaluation* (Subtask 1). The goal of the task is to benchmark Arabic language models on their ability to answer culturally grounded multiple-choice questions in Modern Standard Arabic (MSA). The questions span various domains such as history, customs, geography, literature, and food, and are designed to reflect general cultural literacy in Arab countries.

Participants are provided with a training and development set of MCQs, each with four answer options. The final evaluation is performed on a held-out test set of 2,000 questions, with accuracy as the primary metric. The task encouraged the use of external data for model enhancement, provided that models remain under 13 billion parameters and final checkpoints are submitted for evaluation.

3.2 Dataset

The *PalmX 2025 Cultural Evaluation* dataset consists of 2,000 training examples and 500 development examples, each formulated as a MCQ with four answer options and a single correct answer. We used the training split for fine-tuning and reserved the development set for evaluation, except for our final iterations, where we use both training and evaluation splits for fine-tuning.

3.3 Data Augmentation

Palm. To complement PalmX dataset, we incorporated the Palm dataset (Alwajih et al., 2025a), a broader community-curated resource created by contributors from the 22 Arab countries. Unlike PalmX, which is entirely in MSA, Palm spans both MSA and various dialects, offering instruction-style QA pairs on 20 culturally relevant topics such as heritage, cuisine, history, and proverbs. All examples are manually written by native speakers with cultural familiarity, ensuring authenticity and regional diversity. Although Palm includes training and test splits, only the test portion, comprising 1,926 QA pairs, is publicly available.

We split the available Palm test set into two halves: one for fine-tuning, and the other for evaluation. The splits were created using stratified sampling based on country, ensuring balanced representation across regions in both halves. To bring its free-form QA format in line with PalmX, we converted each example into MCQ format using GPT-4.1. Specifically, we generated three plausible distractors per question, preserving semantic coherence and cultural plausibility. In Appendix 3, we provided the prompt that we used for MCQ version of the palm dataset.

Extending PalmX Dataset. To further diversify and expand our training data, we leveraged the NativQA framework (Alam et al., 2025) in combination with GPT-4.1. The NativQA framework can seamlessly curate large-scale QA pairs based on user queries, ensuring cultural and regional alignment in native languages. GPT-4.1 was selected for its optimal trade-off between cost and performance at the time of experimentation. In all cases, we employed zero-shot prompting with GPT-4.1.

As illustrated in Figure 1, our process for extending the PalmX dataset began by identifying the country associated with each question using GPT-4.1. The prompt used for this task is provided in Listing 3. This country information was then combined with the NativQA framework to curate location-specific QA pairs.

The NativQA framework's retrieval process was carried out in two iterations to maximize topical diversity. To maintain factual quality, all answers were filtered using NativQA's Domain Reliability Check (DRC), which retains only those sourced from NativQA-verified web domains. Furthermore, GPT-4.1 was employed to filter and refine the answers described in Listing 1. The idea is to remove culturally irrelevant or factually incorrect QA pairs and refine answers for conciseness and the overall quality of the dataset. Similar to the Palm test set, these new entries were converted into MCQ format to match PalmX, using the same prompt applied to the original Palm data. This process augmented the original dataset with culturally rich examples while preserving structural and contextual consistency with the PalmX questions. We also manually reviewed 50 samples, which received an average score of about 7.4 on a scale of 10 for accuracy and clarity. We refer to this dataset as the PalmX-ext set. In Table 1, we report the distribution of the dataset that we used for training and evaluation.

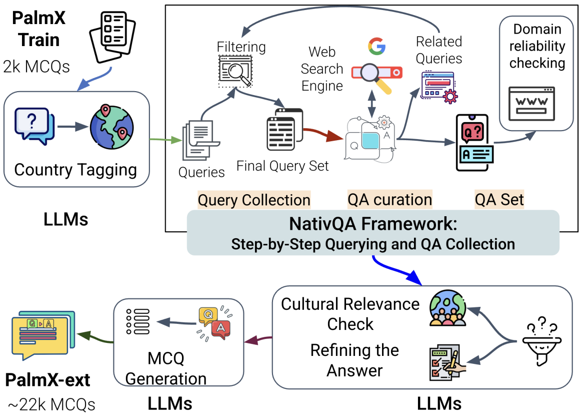


Figure 1: Pipeline for extending the PalmX dataset using the NativQA framework and GPT-4.1.

Data	Train	Dev	Test
PalmX	2,000	500	2,000
Palm	950	950	-
PalmX-ext	22,000	-	-

Table 1: Distribution of the datasets used for training, development and test.

4 Experiments

Models. We began by evaluating a set of open-sourced LLMs in a zero-shot setup on both evaluation datasets. This initial comparison helped us identify the most promising model for fine-tuning, and further demonstrated the utility of the Palm test set as an effective evaluation benchmark.

To identify the most suitable model for the task, we evaluated a set of models based on their performance on the PalmX development set. The models for experiments include `tiny-random-LlamaForCausalLM`,[2] `Qwen2.5-7B-Instruct` (Wang et al., 2024), `Jais-13B-Chat` (Sengupta et al., 2023), `Miraj Mini`,[3] `Llama-3.1-8B-Instruct` (Touvron et al., 2023), `NileChat-3B` (Mekki et al., 2025), `ALLaM-7B-Instruct` (Bari et al., 2025), and `Fanar-7B-Instruct` (Team et al., 2025). We selected both Arabic-centric and multilingual models to compare the effectiveness of models tailored to Arabic with those trained on broader multilingual corpora. The `tiny-random-LlamaForCausalLM` model was included for baseline results.

Training Setup. We experimented with two fine-tuning approaches: LoRA and QLoRA. LoRA trained only a set of low-rank adapter layers while keeping the rest of the model frozen, whereas

[2] https://huggingface.co/HuggingFaceH4/tiny-random-LlamaForCausalLM

[3] https://huggingface.co/arcee-ai/Meraj-Mini

QLoRA combined 4-bit quantization with LoRA adapters to reduce memory usage without a substantial drop in performance.

Both methods were trained on the same mix of datasets: PalmX Train, Palm, and PalmX-ext. We used Fanar's native tokenizer with its default tokenization strategy, a batch size of 4, and gradient accumulation steps of 4. Training was conducted for 3 epochs with a learning rate of 2×10^{-4}, saving the best-performing checkpoint at the end of each run. Fine-tuning followed a specific prompt - each question was prefixed by a system prompt, followed by the question and answer choices, as shown in Figure 2.

Data Augmentation. We then studied the effect of our data augmentation strategy by comparing LoRA training on **PalmX Train** alone *vs.* LoRA training on **PalmX Train** combined with our augmented **Palm** and **PalmX-ext** datasets. This experiment used the same configuration as the earlier LoRA *vs.* QLoRA comparison. After identifying the best-performing approach, we performed hyperparameter tuning to optimise its performance. In Appendix B and C, we report complete experimental setup and results, respectively.

After identifying the best-performing model, the most effective fine-tuning strategy, and the optimal hyperparameters, the final submission was trained for 3 epochs with a learning rate of 2×10^{-4}, LoRA rank 64, dropout 0.1, and scaling factor $\alpha = 16$, using PalmX Train and Dev, Palm, and PalmX-ext.

5 Results

Zero-shot Performance. Table 2 reports the zero-shot performance of several multilingual and Arabic-centric instruction models. **Fanar-7B** achieved the highest accuracy on PalmX Dev, making it our choice for fine-tuning.

Model	PalmX Dev	Palm
tiny-random-Llama	23.40	26.51
Qwen2.5-7B-Inst.	69.20	74.32
Jais-13B-chat	61.00	55.72
Miraj Mini	70.20	**75.99**
Llama3.1-8B-Inst.	66.60	74.06
Nilechat-3B	70.00	66.89
ALLaM-7B-Inst.	70.60	74.32
Fanar-7B	**72.40**	73.34

Table 2: Zero-shot performance of base models.

Comparison on PEFT methods. Table 3 compares LoRA with its quantized variant (QLoRA) under identical settings. LoRA achieved a slight improvement over QLoRA on PalmX Dev, suggesting that full-precision adapters were marginally more effective.

Method	PalmX Dev (%)
QLoRA (4-bit)	80.00
LoRA	**80.60**

Table 3: Results using PEFT methods.

Effect of Data Augmentation. Table 4 evaluates the impact of adding augmented Palm and PalmX-ext data to PalmX Train. The augmented dataset led to substantial gains on PalmX Dev, indicating improved generalization. A more detailed error analysis is provided in Appendix D.

Training Data	PalmX Dev (%)
PalmX	76.6
PalmX + PalmX-ext + Palm	**80.6**

Table 4: Results with and without augmented data.

The final submitted model achieved an accuracy of 84.1% on the Palm test set. As the PalmX development set was included in the training data, it was excluded from evaluation on the submitted model. On the blind test set, the model obtained an accuracy of 70.5%. A more detailed analysis of the discrepancy between Dev and Test performance is provided in Appendix E.

6 Conclusions and Future Work

In this paper, we present our system, *CultranAI*, designed to enhance cultural knowledge representation in LLMs for Arabic. We conduct an extensive comparative evaluation in a zero-shot setting using various multilingual and Arabic-centric models, which led us to identify *Fanar* as the most suitable model for further experimentation. To assess performance in low-compute scenarios, we explored different PEFT methods. We also investigated data augmentation techniques aimed at improving model accuracy. Our proposed system achieved an accuracy of 84.1% on the Palm set, and ranked 5^{th} on the blind test set with an accuracy of 70.5%. Future work will focus on refining data augmentation pipelines and further exploring model generalizability.

7 Limitations

While augmentation brought clear improvements, we believe the performance could have been higher with more careful dataset preparation. In the Palm dataset, instructional QAs were directly converted to MCQ, but some QAs exceeded the 512-token PalmX limit. PalmX-ext avoided this by reformatting MCQs in the first post-processing step. Another problem was distractor quality: in both PalmX-ext and Palm, distractors were often shorter than the correct answer. These issues can be addressed by refining the prompts for distractor generation and adding a processing step to truncate long Palm QAs.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Samir Abdaljalil, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, Nizi Nazar, Youssef Elshahawy, Ahmed Ali, Nadir Durrani, Natasa Milic-Frayling, and Firoj Alam. 2024. LAraBench: Benchmarking Arabic AI with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–520, St. Julian's, Malta. Association for Computational Linguistics.

Rawan Al-Matham, Kareem Darwish, Raghad Al-Rasheed, Waad Alshammari, Muneera Alhoshan, Amal Almazrua, Asma Al Wazrah, Mais Alheraki, Firoj Alam, Preslav Nakov, Norah Alzahrani, Eman alBilali, Nizar Habash, Abdelrahman El-Sheikh, Muhammad Elmallah, Haonan Li, Hamdy Mubarak, Mohamed Anwar, Zaid Alyafeai, and 24 others. 2025. BALSAM: A platform for benchmarking arabic large language models. *arXiv preprint arXiv:2507.22603*.

Firoj Alam, Md Arid Hasan, Sahinur Rahman Laskar, Mucahid Kutlu, and Shammur Absar Chowdhury. 2025. NativQA Framework: Enabling llms with native, local, and everyday knowledge. *arXiv preprint arXiv:2504.05995*.

Fakhraddin Alwajih, Abdellah El Mekki, Samar Mohamed Magdy, AbdelRahim A. Elmadany, Omer Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, Abdulfattah Mohammed Yahya, Rahaf Alhamouri, Hamzah A. Alsayadi, Hiba Zayed, Sara Shatnawi, Serry Sibaee, Yasir Ech-chammakhy, Walid Al-Dhabyani, Marwa Mohamed Ali, Imen Jarraya, and 25 others. 2025a. Palm: A culturally inclusive and linguistically diverse dataset for Arabic LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32871–32894, Vienna, Austria. Association for Computational Linguistics.

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025b. PalmX 2025: The first shared task on benchmarking llms on arabic and islamic culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Lama Ayash, Hassan Alhuzali, Ashwag Alasmari, and Sultan Aloufi. 2025. Saudiculture: A benchmark for evaluating large language models' cultural competence within saudi arabia. *Journal of King Saud University Computer and Information Sciences*, 37(6):123.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham Abdullah Alyahya, Sultan AlRashed, Faisal Abdulrahman Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Saad Amin Hassan, Dr. Majed Alrubaian, Ali Alammari, Zaki Alawami, and 7 others. 2025. ALLam: Large language models for arabic and english. In *The Thirteenth International Conference on Learning Representations*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. Technical report, Microsoft Research.

Yu Ying Chiu, Liwei Jiang, Bill Yuchen Lin, Chan Young Park, Shuyue Stella Li, Sahithya Ravi, Mehar Bhatia, Maria Antoniak, Yulia Tsvetkov, Vered Shwartz, and Yejin Choi. 2025. CulturalBench: A robust, diverse and challenging benchmark for measuring LMs' cultural knowledge through human-AI red-teaming. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25663–25701, Vienna, Austria. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Md. Arid Hasan, Maram Hasanain, Fatema Ahmad, Sahinur Rahman Laskar, Sunaya Upadhyay, Vrunda N Sukhadia, Mucahid Kutlu, Shammur Absar Chowdhury, and Firoj Alam. 2025. NativQA: Multilingual culturally-aligned natural query for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14886–14909, Vienna, Austria. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. GooAQ: Open question answering with diverse answer types. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 421–433, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024a. CultureLLM:: Incorporating cultural differences into large language models. *Advances in Neural Information Processing Systems*, 37:84799–84838.

Cheng Li, Damien Teney, Linyi Yang, Qingsong Wen, Xing Xie, and Jindong Wang. 2024b. CulturePark: Boosting cross-cultural understanding in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 65183–65216.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities. *arXiv preprint arXiv:2505.18383*.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2025. AraDiCE: Benchmarks for dialectal and cultural capabilities in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4186–4218, Abu Dhabi, UAE. Association for Computational Linguistics.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, and 1 others. 2024. BLEnD: A benchmark for llms on everyday knowledge in diverse cultures and languages. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2025. Survey of cultural awareness in language models: Text and beyond. *Computational Linguistics*, pages 1–96.

Rifki Putri, Faiz Haznitrama, Dea Adhista, and Alice Oh. 2024. Can llm generate culturally relevant commonsense qa data? case study in indonesian and sundanese. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20571–20590.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, and 1 others. 2023. Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Weiyan Shi, Ryan Li, Yutong Zhang, Caleb Ziems, Sunny Yu, Raya Horesh, Rogério Abreu De Paula, and Diyi Yang. 2024. CultureBank: An online community-driven knowledge base towards culturally aware language technologies. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4996–5025, Miami, Florida, USA. Association for Computational Linguistics.

Fanar Team, 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, and 23 others. 2025. Fanar: An arabic-centric multimodal generative ai platform.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

A Prompts

<bos> You're a helpful Arabic assistant that answers multiple-choice questions accurately. Choose the best answer based only on the given question and options.
<start_of_turn>user
السؤال
A.
الخيار الأول
B.
الخيار الثاني
C.
الخيار الثالث
D.
الخيار الرابع

<end_of_turn>
<start_of_turn>model
Answer <end_of_turn>

Figure 2: Example of a formatted prompt used for Arabic MCQ fine-tuning.

```
system_prompt = """
You are an advanced NLP annotation assistant
specializing in evaluating Arabic questions and
answers. Your role is to classify questions,
assess answers, and refine them for conciseness
and accuracy.

Follow the structured guidelines for
classification:
- **Step 1: Evaluate and refine the answer**,
ensuring it is concise and factually correct.
- **Step 2: Determine if the question-answer
pair is relevant to the Arabic culture.

### **Annotation Task**
You are an expert Arabic NLP QA annotator. Your
task is to evaluate and refine a
question-answer pair based on the following
steps:

### **Step 1: Evaluate and Edit the Answer**
- **Answer Evaluation:**
  - **Correct:** Fully and accurately answers
the question.
  - **Incorrect:** Does not answer the question
or contains false information.
  - **Partially Correct:** Provides some
relevant information but is incomplete.
- **Answer Refinement:**
  - If correct or partially correct but **too
long, vague, or redundant**, rewrite it to be
**concise and precise**.

### **Step 2: Determine Arabic cultural
relevance**
- **Yes:** The question explicitly refers to
the Arabic culture.
- **No:** The question is about a different
culture than Arabic.
- **Unsure:** It is difficult to determine
whether the question refers to any specific
culture.
"""

user_prompt = f"""
### **Input Data:**

Question: {data['question']}
Answer: {data['answer']}

### **Your Response in JSON format:**
{
"answer_evaluation": "Correct" or "Incorrect"
or "Partially Correct",
"corrected_answer": "Provide a concise, precise
answer if needed, otherwise leave empty.",
"culture_relevance": "Yes" or "No" or "Unsure"
}
"""
```

Listing 1: Prompt for evaluating, refining, and filtering Arabic QA pairs.

```
system_prompt = """
You are an expert in educational content
creation specializing in Arabic language and
culture. Your task is to convert culturally
relevant question-answer pairs into
multiple-choice questions (MCQs) by generating
three plausible, culturally relevant, and
contextually appropriate incorrect answer
options (distractors) in Arabic for each
question.


Requirements:
- All options must be in Arabic.
- Distractors must be plausible and relevant to
the question.
- Avoid answers that are obviously incorrect,
unrelated, or closely paraphrase the correct
answer.
- Output only the 3 incorrect answers in the
following JSON format:

JSON Output format:
{{
"A.": "",
"B": "",
"C": ""
}}
"""

user_prompt = f"""
Given the following question and its correct
answer, generate 3 plausible but incorrect
answer options in Arabic.

Question: "{data['question']}"
Correct Answer: "{data['answer']}"
"""
```

Listing 2: Prompt for generating 3 plausible distractors.

```
system_prompt = "You are an AI assistant for
country identification."

user_prompt = """
You are an expert in Arab culture and geography.
Given a question in Arabic, your task is to
identify the most relevant Arab
country that the question is likely referring
to, either explicitly or implicitly.

Always return the name of a single Arab country
in English
(e.g., Qatar, Egypt, Saudi Arabia, UAE,
Morocco, etc.).

Even if the country is not directly named, use
cultural, linguistic,
environmental, or historical clues to infer the
closest matching Arab country.

Return your response in JSON format with a
single field "country"
containing only the country name.

QUESTION: "{question}"
"""
```

Listing 3: Prompt for identifying country.

Epochs	LR	r	Dropout	Alpha	PalmX Dev (%)
4	5e-5	64	0.15	16	79.6
5	1.2e-4	64	0.05	16	79.8
3	5e-5	64	0.05	16	79.8
4	5e-5	64	0.10	16	80.2
5	1e-4	32	0.05	32	80.4
3	2e-4	64	0.05	16	**80.5**

Table 5: PalmX Dev results from hyperparameter tuning.

B Hyperparameters

Hyperparameter tuning varied the number of epochs (3–5), the learning rates (5×10^{-5} to 2×10^{-4}), the dropout rates (0.05, 0.1, 0.15), and the LoRA-specific parameters such as the rank ($r = 32$ or 64) and the scaling factor ($\alpha = 16$ or 32). Starting from a baseline, we tested higher epochs, lower learning rates, and increased dropout for regularization effects, as well as a reduced-rank, higher-α variant ($r = 32$, $\alpha = 32$). Each configuration was trained and evaluated on the PalmX Dev set to ensure consistency in reporting.

C Results on the Hyperparameter Tuning

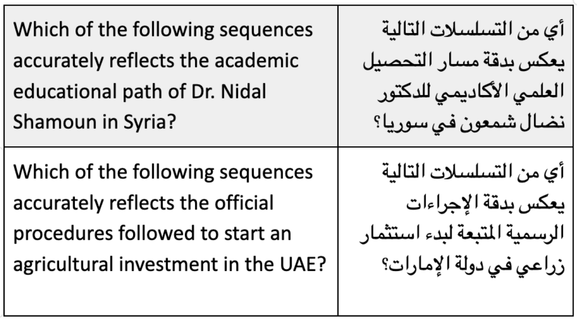
Fine-tuning experiments with Fanar-7B are summarized in Table 5. The top setup used 3 epochs, a 2×10^{-4} learning rate, LoRA rank 64, dropout 0.05, and $\alpha = 16$, yielding an average accuracy of 80.5 on PalmX Dev. We also observed a slight improvement when increasing the dropout to 0.1 in an earlier run with a similar configuration, and therefore incorporated this change into the top-performing setup to form our final configuration.

D Error Analysis: Effect of Augmentation

To better understand the impact of augmentation, we analyzed the subset of questions from the PalmX 2025 development set that the base model (Fanar-9B-Instruct) failed to answer correctly. Out of 500 questions, Fanar produced 138 errors.

Finetuning on PalmX alone corrected 38 of these errors. When augmented data was included, the model solved an additional 53 questions, while losing accuracy on only 3 of the 38 cases previously resolved. In total, the augmented model recovered 88 of the 138 initially incorrect items.

Representative examples of these improvements are shown in Figures 3 and 4. These illustrate how augmentation introduced broader topical coverage, especially on less-documented cultural and regional details. Without augmentation, the model remained limited to narrower knowledge encoded in PalmX.

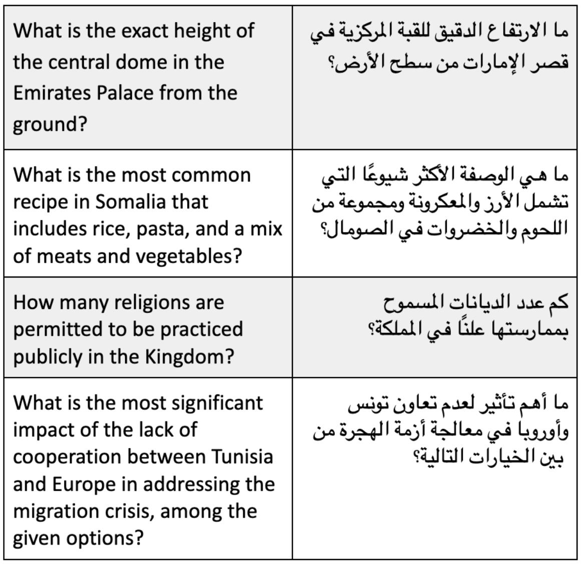


Which of the following sequences accurately reflects the academic educational path of Dr. Nidal Shamoun in Syria?	أي من التسلسلات التالية يعكس بدقة مسار التحصيل العلمي الأكاديمي للدكتور نضال شمعون في سوريا؟
Which of the following sequences accurately reflects the official procedures followed to start an agricultural investment in the UAE?	أي من التسلسلات التالية يعكس بدقة الإجراءات الرسمية المتبعة لبدء استثمار زراعي في دولة الإمارات؟

Figure 3: Questions solved by both PalmX-only and Augmentation.

E Error Analysis: Dev vs. Test Performance

We also examined the discrepancy between the Dev and Test set performance. While our model showed strong results on Dev, its accuracy dropped considerably on Test. To better understand this, we compared representative samples of questions from Train, Dev, and Test.



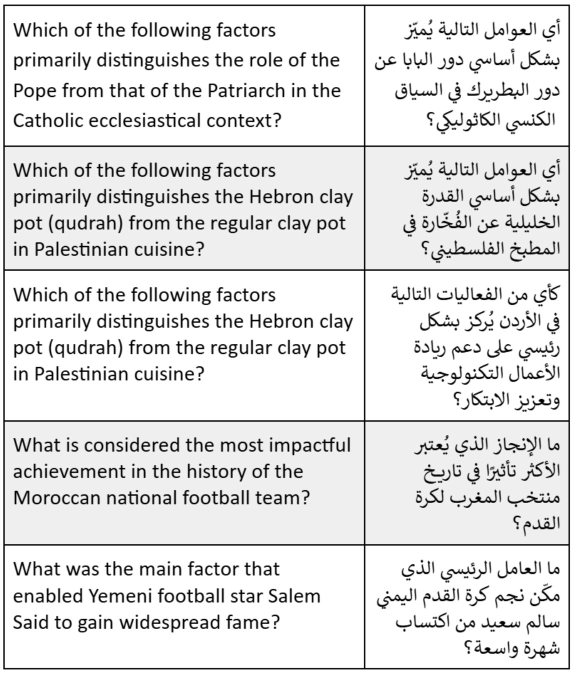
What is the exact height of the central dome in the Emirates Palace from the ground?	ما الارتفاع الدقيق للقبة المركزية في قصر الإمارات من سطح الأرض؟
What is the most common recipe in Somalia that includes rice, pasta, and a mix of meats and vegetables?	ما هي الوصفة الأكثر شيوعًا التي تشمل الأرز والمعكرونة ومجموعة من اللحوم والخضروات في الصومال؟
How many religions are permitted to be practiced publicly in the Kingdom?	كم عدد الديانات المسموح بممارستها علنًا في المملكة؟
What is the most significant impact of the lack of cooperation between Tunisia and Europe in addressing the migration crisis, among the given options?	ما أهم تأثير لعدم تعاون تونس وأوروبا في معالجة أزمة الهجرة من بين الخيارات التالية؟

Figure 4: Questions solved only with Augmentation.

The train and dev sets are closely aligned, focusing on contemporary cultural, institutional, and social knowledge (see Figures 5 and 6). This alignment explains the stronger performance on dev: the model is effectively evaluated on material resembling what it was trained on.

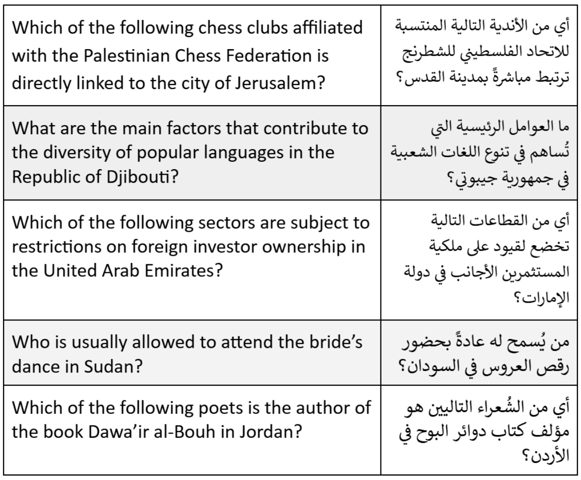
By contrast, the test set introduces broader and less-represented domains, including ancient history, proverbs, zoology, and legal systems (Figure 7). These require background knowledge beyond the distribution covered in training, explaining the observed performance drop.

It should also be noted that model development and checkpoint selection relied on dev, while the test set remained hidden, reinforcing the discrepancy.



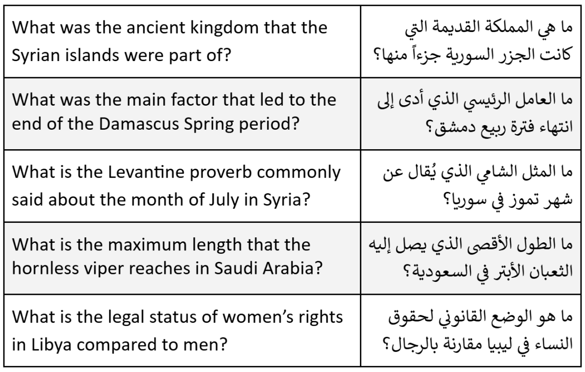
Which of the following factors primarily distinguishes the role of the Pope from that of the Patriarch in the Catholic ecclesiastical context?	أي العوامل التالية يُميّز بشكل أساسي دور البابا عن دور البطريرك في السياق الكنسي الكاثوليكي؟
Which of the following factors primarily distinguishes the Hebron clay pot (qudrah) from the regular clay pot in Palestinian cuisine?	أي العوامل التالية يُميّز بشكل أساسي القدرة الخليلية عن الفُخّارة في المطبخ الفلسطيني؟
Which of the following factors primarily distinguishes the Hebron clay pot (qudrah) from the regular clay pot in Palestinian cuisine?	كأي من الفعاليات التالية في الأردن يُركز بشكل رئيسي على دعم ريادة الأعمال التكنولوجية وتعزيز الابتكار؟
What is considered the most impactful achievement in the history of the Moroccan national football team?	ما الإنجاز الذي يُعتبر الأكثر تأثيرًا في تاريخ منتخب المغرب لكرة القدم؟
What was the main factor that enabled Yemeni football star Salem Said to gain widespread fame?	ما العامل الرئيسي الذي مكّن نجم كرة القدم اليمني سالم سعيد من اكتساب شهرة واسعة؟

Figure 5: Examples from PalmX Cultural Train Set.



Which of the following chess clubs affiliated with the Palestinian Chess Federation is directly linked to the city of Jerusalem?	أي من الأندية التالية المنتسبة للاتحاد الفلسطيني للشطرنج ترتبط مباشرةً بمدينة القدس؟
What are the main factors that contribute to the diversity of popular languages in the Republic of Djibouti?	ما العوامل الرئيسية التي تُساهم في تنوع اللغات الشعبية في جمهورية جيبوتي؟
Which of the following sectors are subject to restrictions on foreign investor ownership in the United Arab Emirates?	أي من القطاعات التالية تخضع لقيود على ملكية المستثمرين الأجانب في دولة الإمارات؟
Who is usually allowed to attend the bride's dance in Sudan?	من يُسمح له عادةً بحضور رقص العروس في السودان؟
Which of the following poets is the author of the book Dawa'ir al-Bouh in Jordan?	أي من الشُعراء التاليين هو مؤلف كتاب دوائر البوح في الأردن؟

Figure 6: Examples from PalmX Cultural Dev Set.



What was the ancient kingdom that the Syrian islands were part of?	ما هي المملكة القديمة التي كانت الجزر السورية جزءاً منها؟
What was the main factor that led to the end of the Damascus Spring period?	ما العامل الرئيسي الذي أدى إلى انتهاء فترة ربيع دمشق؟
What is the Levantine proverb commonly said about the month of July in Syria?	ما المثل الشامي الذي يُقال عن شهر تموز في سوريا؟
What is the maximum length that the hornless viper reaches in Saudi Arabia?	ما الطول الأقصى الذي يصل إليه الثعبان الأبتر في السعودية؟
What is the legal status of women's rights in Libya compared to men?	ما هو الوضع القانوني لحقوق النساء في ليبيا مقارنة بالرجال؟

Figure 7: Examples from PalmX Cultural Test Set.

MarsadLab at PalmX Shared Task: An LLM Benchmark for Arabic Culture and Islamic Civilization

Md. Rafiul Biswas[1], Shimaa Ibrahim[2], Kais Attia[3], Firoj Alam[4], Wajdi Zaghouani[2]
[1]Hamad Bin Khalifa University, Qatar, [2]Northwestern University in Qatar, Qatar
[3]Independent Researcher, Tunisia, [4]Qatar Computing Research Institute, Qatar
{mbiswas,fialm}@hbku.edu.qa, wajdi.zaghouani@northwestern.edu

Abstract

This paper presents our submission to the PalmX 2025 Shared Task on Arabic cultural and religious knowledge comprehension. We focus on training large language models capable of representing domain-specific cultural and religious knowledge in Arabic. Our approach leverages parameter-efficient fine-tuning of the instruction-tuned Qwen2.5-7B model using Low-Rank Adaptation (LoRA). To address the challenges of limited training data, we apply quantization-aware fine-tuning with 4-bit precision, enabling efficient adaptation under constrained resources. The model is further aligned with the multiple-choice evaluation format to enhance task-specific reasoning. Without relying on external data augmentation, our system achieves competitive performance across both the *Arabic General Culture* and *Islamic Culture* subtasks, demonstrating the effectiveness of targeted fine-tuning for enriching cultural and religious knowledge representation in LLMs. On the blind test sets, our systems ranked 7^{th} and 4^{th} in the cultural and Islamic subtasks, respectively. To ensure reproducibility, we make our full codebase and experimental configurations available at https://github.com/rafiulbiswas/PalmX.

1 Introduction

Culturally aware language technologies are essential for high-stakes applications—education, public services, healthcare, and content moderation—where responses must be accurate, respectful, and contextually appropriate. In Arabic settings, a lack of cultural and religious grounding can lead to biased or inappropriate outputs, partly due to the predominance of Western-centric training data in large language models (LLMs) (Ayash et al., 2025; Alwajih et al., 2025b). Addressing this gap requires models that can represent and reason over Arabic cultural heritage and Islamic knowledge, as well as standardized evaluations that make such competence measurable (Sadallah et al., 2025a).

To advance cultural and islamic capabilities in Arabic-centric LLMs PalmX 2025 shared task offered two subtasks—*General Culture* and *Islamic Culture*—using multiple-choice (MCQ) datasets in Modern Standard Arabic (MSA) (Alwajih et al., 2025a). These subtasks probe models' ability to reason about customs, cuisine, history, and Islamic practices, providing a focused testbed for culturally grounded reasoning in Arabic.

Developing such capabilities is challenging. Beyond data imbalance, Arabic presents diglossia, rich morphology, and strong context dependence, all of which complicate knowledge representation and question answering (Hasan et al., 2025). Practical constraints—limited labeled data and domain-specific MCQ formats—further motivate resource-efficient adaptation strategies.

We adapt an instruction-following LLM to these subtasks using parameter-efficient fine-tuning. Concretely, we fine-tune the 7B-parameter Qwen2.5-Instruct (Team, 2025) with Low-Rank Adaptation (LoRA) (Hu et al., 2022) on the official PalmX training sets (Alwajih et al., 2025a), enabling effective domain adaptation under modest compute. At inference, we employ prompt-based strategies to inject expert priors and enforce output constraints (e.g., instructing the model to act as an "expert in Arabic culture and Islamic studies" and to output only the option letter). Empirically, careful prompt design yields consistent but modest gains in MCQ accuracy; closing the remaining gap will likely require richer cultural grounding and more structured supervision. To summarize, our contributions include:

- We adapt Qwen2.5-7B-Instruct to Arabic cultural and religious knowledge using Low-Rank Adaptation (LoRA) with 4-bit quantization-aware fine-tuning, achieving effective domain specialization under modest computational budgets.
- We introduce inference-time instruction templates and output-space constraints that align the

Proceedings of The Third Arabic Natural Language Processing Conference, pages 818–824
November 8-9, 2025 ©2025 Association for Computational Linguistics

model with the multiple-choice setting (expert prior + option-letter output), yielding consistent accuracy gains without additional supervision.

- Our system attains resonable performances on PalmX 2025 *General Culture* and *Islamic Culture*, ranking 7^{th} and 4^{th} on the blind test sets, respectively, without recourse to external data augmentation.
- We provide a concise pipeline demonstrating that low-compute PEFT can reliably enrich cultural/religious knowledge in Arabic LLMs.

2 Related Works

Benchmarking language models for Arabic has progressed along two complementary lines: inclusion within multilingual suites and dedicated evaluations of large language models (LLMs) for Arabic. Early efforts commonly incorporated Arabic into broad benchmarks such as XGLUE, XTREME, XTREME-R, GEM, and Dolphin, covering a spectrum of tasks that emphasized classification (e.g., natural language inference), sequence labeling (part-of-speech tagging, named entity recognition), and generation (summarization) (Liang et al., 2020; Hu et al., 2020; Ruder et al., 2021; Gehrmann et al., 2021; Nagoudi et al., 2023). More recent work has turned to Arabic-focused LLM assessment, evaluating standard and Arabic-centric models on task suites and datasets (Sengupta et al., 2023; Khondaker et al., 2023; Abdelali et al., 2024; Dalvi et al., 2024), probing the effects of prompting in native (Arabic) versus non-native (English) languages (Kmainasi et al., 2025), and extending analyses to multimodal settings (Alwajih et al., 2024; Das et al., 2024).

Within cultural evaluation, prior studies quantify representational bias in entity mentions toward Western versus Arab contexts (Naous et al., 2024), assess cultural alignment using constructs from the World Values Survey (AlKhamissi et al., 2024), and introduce culture-aware diagnostic and QA resources (Arora et al., 2024; Myung et al., 2024; Alam et al., 2025). Complementing these efforts, Arabic-focused benchmarks have begun to appear: ARADICE targets dialect comprehension and cultural QA (Mousi et al., 2024), while other resources probe cultural values and regional knowledge via translated survey instruments and Wikipedia-derived questions (Al-Matham et al., 2025). Despite these advances, converging evidence indicates that general-purpose LLMs still underperform on culturally grounded reasoning and Arabic commonsense, underscoring the need for benchmarks, resources, and model-development methods explicitly tailored to Arabic cultural and dialectal contexts (Sadallah et al., 2025b; Yakhni and Chehab, 2025; Qian et al., 2024).

PalmX 2025 (Alwajih et al., 2025a) advances this research area with a curated, competition-driven evaluation of Arabic cultural capabilities in Modern Standard Arabic, spanning *General Culture* and *Islamic Culture*. QAs are designed to cover all Arab countries and key Islamic concepts, providing a focused MCQ testbed and strong baselines (e.g., *NileChat-3B*) (Mekki et al., 2025; Alwajih et al., 2025b). Our work aligns with this direction by adapting an instruction-tuned LLM to PalmX via parameter-efficient fine-tuning and expert-persona prompting, and by analyzing remaining performance gaps relative to culturally trained baselines.

3 Dataset

PalmX 2025 provides two Modern Standard Arabic (MSA) multiple-choice (four-option) datasets that target complementary facets of cultural knowledge.

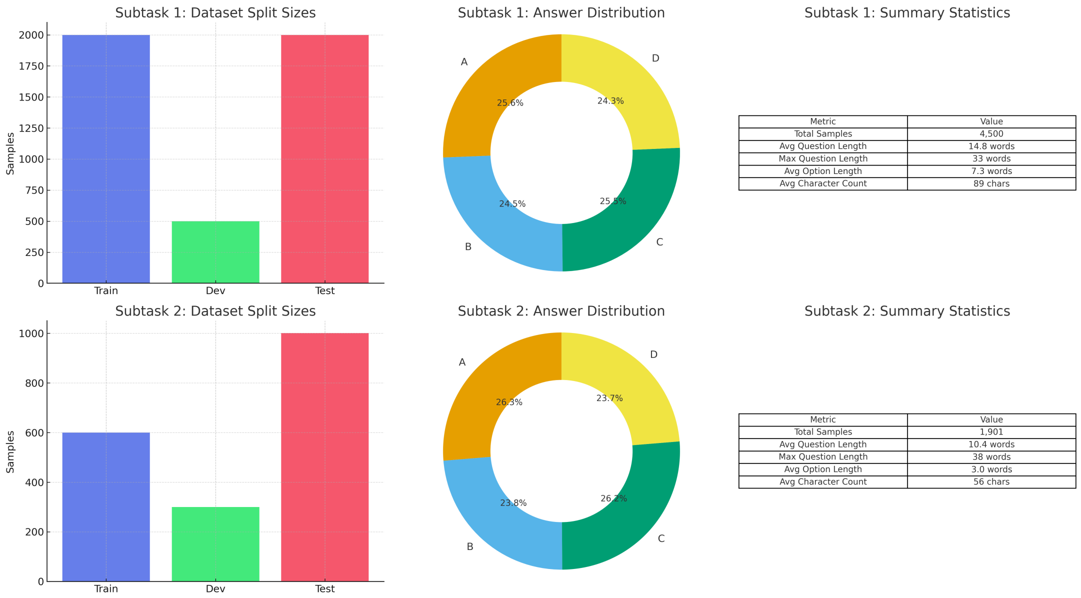
Task 1: General Culture comprises 4,500 questions spanning Arab culture across 22 countries, with official splits of 2,000 training, 500 development, and 2,000 test items.

Task 2: Islamic Culture contains 1,900 questions focused on Islamic cultural knowledge, split into 600 training, 300 development, and 1,000 test items. All experiments in this work use the organizers' official splits without external data augmentation.

Both subtasks follow a consistent data distribution structure, previously unseen questions for blind testing, with accuracy serving as the primary evaluation metric (see in figure 1).

4 System Overview

We experimented with different open sources LLM such as NileChat-3B (Mekki et al., 2025), LLaMA3.1 8B (Touvron et al., 2023), Fanar-1-9B-Instruct (Team et al.) and Qwen2.5-7B-Instruct (Team, 2025). Qwen2.5-7B-Instruct outperformed over other LLM and so we adapt Qwen2.5-7B-Instruct to Arabic cultural understanding via parameter-efficient fine-tuning with Low-Rank Adaptation (LoRA) (Hu et al., 2022).



Metric	Value
Total Samples	4,500
Avg Question Length	14.8 words
Max Question Length	33 words
Avg Option Length	7.3 words
Avg Character Count	89 chars

Metric	Value
Total Samples	1,901
Avg Question Length	10.4 words
Max Question Length	38 words
Avg Option Length	3.0 words
Avg Character Count	56 chars

Figure 1: Dataset statistics in two subtasks.

4.1 Training Methodology

Prompting and supervision: We formatted training examples using a structured instruction-following template for Arabic cultural question-answering. Each instance comprises a system message, the user turn containing the question and the four labeled options, and an assistant turn with *only* the correct option letter. We implement this using the model's native chat template markers (`<|im_start|>` / `<|im_end|>`) to delimit turns. Explicitly constraining the target to the option letter suppresses verbosity, improves label consistency, and simplifies answer extraction at evaluation time; supervision is via standard next-token cross-entropy over the assistant turn.

Optimization setup: We train for three epochs (selected via development-set performance) with a learning rate of 2×10^{-4} and a linear warmup of 100 steps. We use an effective batch size of 16 via per-device batch size $= 4$ and gradient accumulation $\times 4$. Mixed precision uses `bfloat16` where supported (falling back to `fp16`), and the maximum sequence length is 512 tokens, which comfortably covers all MCQ contexts in our data.

Memory efficiency: To enable fine-tuning on commodity GPUs, we combine 4-bit NF4 quantization of the base weights with gradient checkpointing, trading additional compute for a reduced activation footprint. In practice, this configuration supports single-GPU training with $\sim$8 GB of memory. During preprocessing, we tokenize in mini-batches (size 100) to avoid holding the entire tokenized corpus in memory, and we periodically release cached CUDA memory to mitigate fragmentation during longer runs.

Multi-task adapters: For the two PalmX subtasks, we train separate LoRA adapters on the same quantized backbone to avoid negative transfer across cultural domains while retaining a unified deployment artifact. The *General Culture* adapter is fine-tuned on $\sim$2,000 instances, and the *Islamic Culture* adapter on $\sim$600 instances. This modular design permits task-specific specialization and lightweight "hot-swapping" at inference time without reloading the base model.

4.2 Ablation Study

To better understand the contribution of different components in our system, we conducted comprehensive ablation experiments examining the impact of LoRA hyperparameters, and prompt engineering choices.

LoRA Rank Analysis: We investigated the effect of LoRA rank on model performance and computational efficiency. Table 1 presents results for different rank configurations while keeping other hyperparameters constant ($\alpha = 32$, dropout=0.1).

The results reveal a clear performance improvement from rank 4 to 16, with diminishing returns beyond rank 16. Our chosen rank of 16 represents the optimal balance.

Target Module Selection: We evaluated different combinations of target modules for LoRA adaptation. Table 2 shows the impact of different mod-

Rank	Task 1 (%)	Task 2 (%)	Params (M)	Time (h)
4	63.2	69.8	5.24	2.1
8	65.8	72.1	10.49	2.4
16	**67.6**	**74.1**	**20.97**	**3.0**
32	67.9	74.3	41.94	3.8

Table 1: Effect of LoRA rank on performance on test dataset

ule combinations. Targeting all projection matrices yields the best performance, with attention modules alone outperforming Feed-Forward Neural (FFN) Network modules, suggesting that adapting attention patterns is more crucial for cultural understanding.

Target Modules	Task 1 (%)	Task 2 (%)
Attention (q, v)	64.3	70.2
Attention (q, k, v, o)	66.1	72.5
FFN only	63.7	69.4
All (Attn + FFN)	**67.6**	**74.1**

Table 2: Performance of different target module configurations

Prompt Engineering Variations: We tested several prompt variations to identify the most effective format for Arabic cultural questions.

Prompt Strategy	Task 1 (%)	Task 2 (%)
English system	64.7	71.2
Arabic system	66.3	72.8
Expert framing	**67.6**	**74.1**
Expert + few-shot	66.9	73.5

Table 3: Effect of prompt engineering strategies

The expert framing prompt that positions the model as "an expert in Arabic culture and Islamic studies" yields the best results. Adding chain-of-thought or few-shot examples slightly decreased performance.

5 Result

In Table 4, we report the performance of different models for both tasks before and after fine-tuning. Across both subtasks, performance varies markedly by model and adaptation strategy. The fine-tuned Qwen2.5-7B-Instruct yields the strongest overall results, attaining (67.55%) accuracy on *Task 1: General Culture* and (74.13%) on *Task 2: Islamic Culture*. Fine-tuning provides substantial improvements for all models except Fanar 7B on Task 2, with gains ranging from 7.8 to 12.2 percentage points on Task 1. Notably, Qwen2.5-7B demonstrates the most consistent improvement, gaining 7.75 points on Task 1 and 8.73 points on Task 2. Relative to the task with top ranked system (72.15%) and (84.22%), respectively, this places our best system within (4.60) percentage points on General Culture and (10.09) points on Islamic Culture, indicating substantial room for improvement, especially for the latter.

A cross-task comparison reveals a general trend of improved accuracy on the Islamic subtask after fine-tuning. For Qwen2.5-7B, the gain from Task 1 to Task 2 is (+6.58) percentage points. The NileChat-3B baseline is comparatively stable at (≈64%) on both tasks after fine-tuning, while Llama 3.1 8B-Instruct exhibits a modest uplift over this baseline on Islamic Culture (about (+4.9) points). An exception to the broader trend is Fanar 7B, which performs competitively on General Culture (66.0%) but declines on Islamic Culture (62.4%) compared to its baseline performance (49.6%), suggesting that while fine-tuning improves its general performance, domain- or data-mismatch effects persist that merit further analysis.

These results demonstrate three key observations. First, parameter-efficient fine-tuning confers clear benefits over off-the-shelf models for culturally grounded question answering in Arabic, with consistent improvements observed across most model-task combinations. Second, the effectiveness of fine-tuning varies by model architecture and task domain, as evidenced by the differential improvements across models. Third, the persistent gap to the subtask best scores—particularly on Islamic Culture—highlights the difficulty of capturing nuanced, domain-specific knowledge and the need for richer supervision and/or targeted knowledge integration beyond instruction tuning alone.

	General Culture		Islamic Culture	
Model	Before fine tuning(%)	After fine tuning(%)	Before fine tuning(%)	After fine tuning(%)
NileChat-3B	52.30	64.50	51.80	64.00
LLaMA3.1 8B	58.40	65.90	61.70	69.20
Fanar 7B	54.20	66.00	49.60	62.40
Qwen2.5L-7B	**59.80**	**67.55**	**65.40**	**74.13**

Table 4: Performance comparison of language models on test dataset before and after parameter-efficient fine-tuning. All scores represent accuracy percentages.

Computational efficiency. Our approach is computationally lightweight: training *Task 1* (2,000 samples) completes in approximately three hours on a single NVIDIA RTX 3090; with 4-bit quantization, fine-tuning fits within 8 GB of GPU memory. At inference, throughput is about ∼2 s per question on GPU and ∼8 s on CPU. The resulting LoRA checkpoint occupies ∼1.2 GB, compared to ∼15 GB for the full model,

6 Error Analysis

To better understand the limitations and failure modes of our fine-tuned models, we conducted a comprehensive error analysis on a stratified sample of 200 incorrect predictions from our best-performing model (**QWEN2.5L-7B**). Our analysis reveals distinct error patterns across tasks: for General Culture, the primary failure modes include factual knowledge gaps (42%), cultural context misunderstanding (28%), and ambiguous question interpretation (18%). For Islamic Culture, errors predominantly stem from religious text interpretation challenges (35%), difficulty handling sectarian variations (24%), and historical timeline confusion (21%).

When comparing errors across tasks, we also observed common problems such as relying on surface-level patterns instead of deeper understanding, showing overconfidence in culturally ambiguous cases, and favoring Western or standardized views over regional cultural perspectives. These findings suggest that while parameter-efficient fine-tuning improves performance, the models still face challenges in handling complex cultural reasoning that requires deeper context and sensitivity to local variations.

7 Conclusion

This paper presented MarsadLab's approach to the PalmX 2025 shared task on Arabic Islamic and Cultural understanding. Through parameter-efficient fine-tuning using LoRA adaptation of Qwen2.5-7B-Instruct, we achieved competitive performance across both tasks. Our work demonstrates that parameter-efficient methods can effectively adapt LLMs for culturally-nuanced tasks without requiring extensive computational resources. By training only 0.27% of model parameters through LoRA while employing 4-bit quantization, we reduced memory requirements by approximately 75% compared to full fine-tuning, making our approach accessible to researchers with limited GPU resources. Future work includes investigating low-compute and minimal-data regimes for such tasks.

Acknowledgments

This study was supported by the grant NPRP14C-0916-210015, awarded by the Qatar Research, Development and Innovation Council (QRDI).

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Samir Abdaljalil, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, Nizi Nazar, Youssef Elshahawy, Ahmed Ali, Nadir Durrani, Natasa Milic-Frayling, and Firoj Alam. 2024. LAraBench: Benchmarking Arabic AI with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–520, St. Julian's, Malta. Association for Computational Linguistics.

Rawan Al-Matham, Kareem Darwish, Raghad Al-Rasheed, Waad Alshammari, Muneera Alhoshan, Amal Almazrua, Asma Al Wazrah, Mais Alheraki, Firoj Alam, Preslav Nakov, and 1 others. 2025. BALSAM: A platform for benchmarking arabic large language models. *arXiv preprint arXiv:2507.22603*.

F. Alam, Md Asif Hasan, S. R. Laskar, M. Kutlu, Kareem Darwish, and S. A. Chowdhury. 2025. NativQA framework: Enabling llms with native, local, and everyday knowledge. *arXiv preprint arXiv:2504.05995*.

Badr AlKhamissi, Muhammad ElNokrashy, Mai Alkhamissi, and Mona Diab. 2024. Investigating cultural alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12404–12422, Bangkok, Thailand. Association for Computational Linguistics.

Fakhraddin Alwajih, Gagan Bhatia, and Muhammad Abdul-Mageed. 2024. Dallah: A dialect-aware multimodal large language model for arabic. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 320–336.

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025a. PalmX 2025: The first shared task on benchmarking LLMs on arabic culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ANLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Fakhraddin Alwajih, Abdellah El Mekki, Samar Mohamed Magdy, Abdelrahim A Elmadany, Omer

Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, Abdulfattah Mohammed Yahya, and 1 others. 2025b. Palm: A culturally inclusive and linguistically diverse dataset for arabic llms. *arXiv preprint arXiv:2503.00151*.

Shane Arora, Marzena Karpinska, Hung-Ting Chen, Ipsita Bhattacharjee, Mohit Iyyer, and Eunsol Choi. 2024. CaLMQA: Exploring culturally specific long-form question answering across 23 languages. *arXiv preprint arXiv:2406.17761*.

Lama Ayash, Hassan Alhuzali, Ashwag Alasmari, and Sultan Aloufi. 2025. Saudiculture: A benchmark for evaluating large language models' cultural competence within saudi arabia. *Journal of King Saud University Computer and Information Sciences*, 37(6):123.

Fahim Dalvi, Maram Hasanain, Sabri Boughorbel, Basel Mousi, Samir Abdaljalil, Nizi Nazar, Ahmed Abdelali, Shammur Absar Chowdhury, Hamdy Mubarak, Ahmed Ali, Majd Hawasly, Nadir Durrani, and Firoj Alam. 2024. LLMeBench: A flexible framework for accelerating LLMs benchmarking. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 214–222, St. Julians, Malta. Association for Computational Linguistics.

Rocktim Das, Simeon Hristov, Haonan Li, Dimitar Dimitrov, Ivan Koychev, and Preslav Nakov. 2024. EXAMS-V: A multi-discipline multilingual multimodal exam benchmark for evaluating vision language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7768–7791, Bangkok, Thailand. Association for Computational Linguistics.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, and 37 others. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Md. Arid Hasan, Maram Hasanain, Fatema Ahmad, Sahinur Rahman Laskar, Sunaya Upadhyay, Vrunda N Sukhadia, Mucahid Kutlu, Shammur Absar Chowdhury, and Firoj Alam. 2025. NativQA: Multilingual culturally-aligned natural query for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14886–14909, Vienna, Austria. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multitask benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. GPTAraEval: A comprehensive evaluation of ChatGPT on Arabic NLP. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 220–247, Singapore. Association for Computational Linguistics.

Mohamed Bayan Kmainasi, Rakif Khan, Ali Ezzat Shahroor, Boushra Bendou, Maram Hasanain, and Firoj Alam. 2025. Native vs non-native language prompting: A comparative analysis. In *Web Information Systems Engineering – WISE 2024*, pages 406–420, Singapore. Springer Nature Singapore.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, and 5 others. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities. *arXiv preprint arXiv:2505.18383*.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2024. Aradice: Benchmarks for dialectal and cultural capabilities in llms. *arXiv preprint arXiv:2409.11404*.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, and 1 others. 2024. BLEnD: A benchmark for llms on everyday knowledge in diverse cultures and languages. *arXiv preprint arXiv:2406.09948*.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, Ahmed El-Shangiti, and Muhammad Abdul-Mageed. 2023. Dolphin: A challenging and diverse benchmark for Arabic NLG. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1404–1422, Singapore. Association for Computational Linguistics.

Tarek Naous, Michael Ryan, Alan Ritter, and Wei Xu. 2024. Having beer after prayer? measuring cultural bias in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16366–16393, Bangkok, Thailand. Association for Computational Linguistics.

Zhaozhi Qian, Faroq Altam, Muhammad Alqurishi, and Riad Souissi. 2024. Cameleval: Advancing culturally aligned arabic language models and benchmarks. *arXiv preprint arXiv:2409.12623*.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A. Sadallah, J. C. Tonga, K. Almubarak, S. Almheiri, F. Atif, C. Qwaider, K. Kadaoui, S. Shatnawi, Y. Alesh, and F. Koto. 2025a. Commonsense reasoning in arab culture. *Preprint*, arXiv:2502.12788.

Abdelrahman Sadallah, Junior Cedric Tonga, Khalid Almubarak, Saeed Almheiri, Farah Atif, Chatrine Qwaider, Karima Kadaoui, Sara Shatnawi, Yaser Alesh, and Fajri Koto. 2025b. Commonsense reasoning in arab culture. *arXiv preprint arXiv:2502.12788*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Fanar Team, 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, and 23 others. Fanar: An arabic-centric multimodal generative ai platform.

Qwen Team. 2025. Qwen2.5-vl.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Silvana Yakhni and Ali Chehab. 2025. Can llms translate cultural nuance in dialects? a case study on lebanese arabic. In *Proceedings of the 1st Workshop on NLP for Languages Using Arabic Script*, pages 114–135.

Appendix

Model Architecture

We employ Qwen2.5-7B-Instruct, a 7.61B-parameter causal LLM comprising 28 transformer layers with Grouped Query Attention (GQA) and a context window of up to 131,072 tokens. For our setting, we use the instruction-tuned variant—optimized to follow complex prompts via supervised fine-tuning and RLHF. To reduce memory footprint, the base model is loaded with `4-bit` NF4 quantization (with double quantization) while retaining `bfloat16` compute through `bitsandbytes`; we enable `k-bit` training using `prepare_model_for_kbit_training`. Tokenization relies on the Qwen tokenizer with right padding; when a pad token is not defined, we map `<pad>` to `<eos>`.

LoRA Adaptation Strategy

To specialize both attention patterns and intermediate representations for culturally grounded reasoning, we attach LoRA adapters to the attention projections `q_proj`, `k_proj`, `v_proj`, and `o_proj`, as well as to the feed-forward projections `gate_proj`, `up_proj`, and `down_proj`. After development-set tuning, we adopt a rank $r = 16$, scaling $\alpha = 32$, dropout $= 0.1$, and no bias, a configuration that yields approximately **20.97M** trainable parameters ($\approx$**0.27%** of the base model). In practice, this supports single-GPU fine-tuning with $\sim$8 GB of memory while preserving sufficient capacity for the target tasks.

Hyperparameters

After empirical evaluation on the development set, we selected the following LoRA hyperparameters:

- **Rank (r)**: 16 - Balancing expressiveness with parameter efficiency
- **Scaling factor** (α): 32 - Controlling the magnitude of LoRA updates
- **Dropout**: 0.1 - Preventing overfitting on the limited training data
- **Bias**: None - Following standard LoRA practice

This configuration results in approximately **20.97M trainable parameters** (0.27% of total model parameters), enabling fine-tuning with only 8GB of GPU memory while maintaining model expressiveness for cultural reasoning tasks.

Star at PalmX 2025 Shared Task: Arabic Cultural Understanding via Targeted Pretraining and Lightweight Fine-tuning

Eman Elrefai[1]
[1]Alexandria University
eman.lotfy.elrefai@gmail.com

Esraa Khaled[2]
[2]Cairo University
esraa.k.fouad@gmail.com

Alhassan Ehab[3]
[3]Minia University
alhassanehab186@gmail.com

Abstract

We present a two-stage framework for enhancing Arabic cultural understanding in small language models, specifically designed for PalmX 2025(Alwajih et al., 2025) Shared Task 1: General Culture Evaluation. Our approach combines continuous pretraining on a culturally-enriched Arabic corpus spanning 10 Arab countries and different cultural domains, followed by supervised fine-tuning on cultural question-answering data. Using Parameter-Efficient Fine-Tuning (PEFT) (Zhang et al., 2025) with LoRA on the Qwen3-4B base model, we achieve 74% accuracy on the development set and 64% on the blind test set, ranking our team ninth in the competition. Our system demonstrates the effectiveness of targeted cultural pretraining for improving Arabic language models' cultural competency while maintaining computational efficiency.

1 Introduction

Arabic cultural understanding represents a critical challenge in natural language processing, as existing large language models often lack the nuanced cultural knowledge necessary to serve Arabic-speaking communities effectively. The PalmX 2025 Shared Task 1 focuses on evaluating models' ability to understand and reason about Arabic cultural concepts, traditions, and knowledge across diverse Arab regions.

Our main system strategy employs a two-stage training paradigm: (1) continuous pretraining on culturally-diverse Arabic content to build foundational cultural knowledge. (2) supervised fine-tuning on structured cultural question-answering data to enhance reasoning capabilities. This approach addresses the fundamental challenge of cultural representation in language models while maintaining computational efficiency through parameter-efficient techniques.

Key findings from our work include achieving competitive performance (74% development accuracy, 64% test accuracy) while using only 4B parameters, demonstrating that targeted cultural pretraining significantly improves performance over baseline models, and identifying that multi-domain cultural coverage is essential for robust cultural understanding. The main challenge discovered was balancing broad cultural coverage with deep domain-specific knowledge within computational constraints.

2 Literature Review

The PalmX 2025 Subtask 1 presents a multiple-choice question-answering challenge focused on Arabic cultural knowledge. The input consists of cultural questions in Modern Standard Arabic (MSA) with four possible answers (A, B, C, D), and the output is the correct answer choice.

Example:

Question:
ما هي العاصمة التاريخية للدولة الأموية؟
Choices:
A.بغداد B.دمشق C.القاهرة D.مكة
Answer: B

2.1 Dataset Details

Our pretraining corpus was constructed from Arabic Wikipedia articles covering 10 Arab countries (Bahrain, Egypt, UAE, Iraq, Kuwait, Jordan, Lebanon, Palestine, Syria, Saudi Arabia) and cultural domains like (Media, Sport, Transport, Healthcare, Education, Religion, Economy, History, Festivals, Tourism).

The coverage was limited to these 10 countries due to data availability and quality constraints: some Arab countries had very limited or incomplete Wikipedia content across the chosen domains, which would have introduced imbalance and sparsity into the corpus. Focusing on countries with richer and more representative cultural data ensured

Proceedings of The Third Arabic Natural Language Processing Conference, pages 825–829
November 8-9, 2025 ©2025 Association for Computational Linguistics

both consistency and reliability of the pretraining resource.

The dataset for instruction fine tuning stage comprises cultural questions covering various aspects of Arab heritage, including history, literature, traditions, geography, and social customs. The training set contains 2,000 examples, with a development set of 500 examples for validation. Questions are formulated in MSA and span knowledge from multiple Arab countries and cultural domains.

2.2 Related Work

Previous work in Arabic NLP has focused primarily on general language understanding tasks like (El Mekki et al., 2025; Bari et al., 2024; Sengupta et al., 2023). Cultural understanding in language models has been explored for various languages (Pawar et al., 2025; Nayak et al., 2024), but limited work exists specifically for Arabic cultural knowledge. Our work bridges this gap by combining cultural corpus pretraining with parameter-efficient fine-tuning (LoRA/PEFT) to enhance cultural awareness in small LMs. To the best of our knowledge, this is the first contribution focusing specifically on Arabic cultural evaluation within the PalmX framework.

3 System Overview

3.1 Architecture

Our system builds upon the Qwen3-4B (Yang et al., 2025) base model, selected for its strong multilingual capabilities and computational efficiency. We employ Low-Rank Adaptation (LoRA)(Singhapoo et al., 2025) for parameter-efficient fine-tuning, enabling effective adaptation while minimizing computational overhead.

The LoRA adaptation is applied to multiple attention and feed-forward layers:

$$h = W_0x + \Delta Wx = W_0x + BAx \qquad (1)$$

where W_0 represents the frozen pre-trained weights, $\Delta W = BA$ is the low-rank adaptation with matrices $B \in R^{d \times r}$ and $A \in R^{r \times d}$, and $r \ll d$ is the rank.

3.2 Two-Stage Training Framework

3.2.1 Stage 1: Cultural Pretraining

We perform continuous pretraining (Tack et al., 2025)on a curated Arabic cultural corpus to inject domain-specific knowledge into the model. The pretraining objective follows the standard causal language modeling loss:

$$\mathcal{L}_{pretrain} = -\sum_{i=1}^{T} \log P(x_i | x_{<i}; \theta) \qquad (2)$$

3.2.2 Stage 2: Supervised Fine-tuning

Following cultural pretraining, we fine-tune the model on the PalmX cultural QA dataset using a chat-based instruction format. The fine-tuning process employs response-only training, where gradients are computed only on assistant responses:

$$\mathcal{L}_{finetune} = -\sum_{i \in \mathcal{R}} \log P(x_i | x_{<i}, context; \theta) \qquad (3)$$

where $\mathcal{R}$ denotes response tokens.

3.3 Cultural Corpus Construction

Our pretraining corpus spans 10 Arab countries (Bahrain, Egypt, UAE, Iraq, Kuwait, Jordan, Lebanon, Palestine, Syria, Saudi Arabia) and cultural domains like (Media, Sport, Transport, Healthcare, Education, Religion , Economy, History , Festivals, Tourism). Articles were systematically collected via Wikipedia API as shown in figure [1] and processed through a comprehensive cleaning pipeline including:

Algorithm 1 Cultural Corpus Processing Pipeline

```
Input: Raw Wikipedia articles $D = \{d_1, d_2, ..., d_n\}$
Initialize: Clean corpus $C = \emptyset$
for each article $d_i$ in $D$ do
    Remove HTML tags and formatting
    Filter by language (Arabic content only)
    Apply deduplication using content hashing
    Chunk into sequences $\leq$ 4096 tokens
    Attach article title as metadata (marker for cultural/contextual grounding)
    $C = C \cup \{processed\_chunks\}$
end for
Return: $C$
```

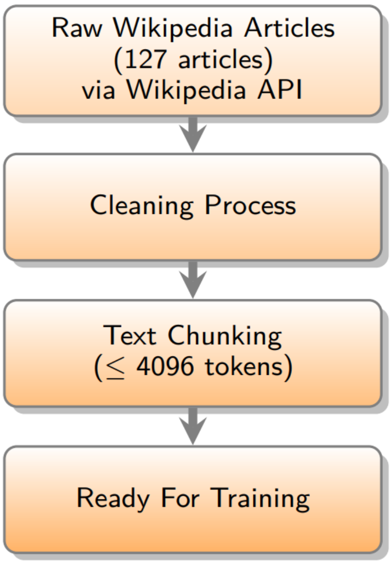


Figure 1: Data Collection Pipeline

4 Experimental Setup

4.1 Data Splits

We utilized the official PalmX dataset splits: 2,000 training examples for supervised fine-tuning, 500 development examples for validation, and a blind test set for final evaluation. For cultural pretraining, we created a development split (3% of cultural corpus) to monitor pretraining progress.The training set contains 4480 examples, with a development set of 139 examples for validation.

4.2 Implementation Details

Our implementation leverages the Unsloth(Han et al., 2023) framework for efficient and scalable training. We summarize the Low-Rank Adaptation (LoRA) configuration and training hyperparameters in table [1].

Parameter	Value
Rank (r) and Alpha	128

Target modules	`q_proj,k_proj`
	`v_proj,o_proj`
	`gate_proj,up_proj`
	`down_proj,lm_head`
	`embed_tokens`

Dropout	0.05
Maximum sequence length	4096

Table 1: LoRA Configuration

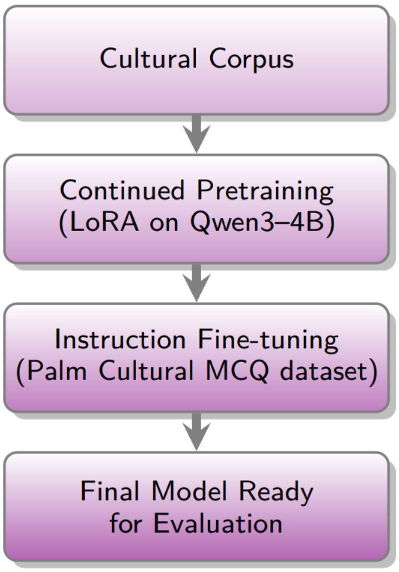


Figure 2: Full Training Pipeline

The configuration adopts a rank and alpha of 128, applies LoRA to multiple attention and projection layers, and supports long-context training with sequences up to 4096 tokens.

The training pipeline consists of two phases: cultural pretraining and cultural QA fine-tuning. Each phase is optimized using the AdamW optimizer with a cosine learning rate schedule and a weight decay of 0.01, with learning rates, epochs, and batch sizes adjusted per phase to balance performance and convergence as shown in table [2].

4.3 Evaluation Metrics

The primary evaluation metric is the accuracy on the MMLU (Nacar et al., 2025). We employ exact match evaluation where the model's predicted letter (A, B, C, D) must exactly match the gold answer. We use MMLU since the competition itself is based on this benchmark because it is widely used to test broad knowledge ability, making it suitable for evaluating general-purpose language models.

4.4 System Pipeline

Our complete training pipeline consists of three sequential stages as shown in figure [2]:

1. **Cultural Pretraining**: Train on cultural corpus for 3 epochs

2. **Cultural QA Fine-tuning**: Train on PalmX cultural dataset for 2 epochs

Task	Hyper- parameters	Other Settings

Cultural Pretraining	`LR: 2e-5,Emb. LR: 5e-6,`
	`Epochs: 3, Batch: 16`
		`AdamW, cosine schedule,`
		`weight-decay:0.01`

Cultural QA Fine-tuning	`LR: 2e-5,`
	`Epochs: 2, Batch: 16`
		`Same as above`

Table 2: Training Hyperparameters

5 Results

5.1 Quantitative Results

Table [3] presents our official evaluation results on the PalmX 2025 Shared Task 1.

Dataset	Accuracy (%)
Development Set	74.0
Blind Test Set	64.0

Table 3: Official evaluation results on PalmX 2025 Subtask 1

5.2 Ablation Studies

We conducted ablation studies to assess the contribution of each training stage:

Configuration	Dev Accuracy (%)
Base Model Only	64.0
Star Model	**74.0**

Table 4: Ablation study showing contribution of each training stage

The results demonstrate that each training stage contributes significantly to final performance, with cultural pretraining providing the largest single improvement (10%) over the base model as shown in table [4].

5.3 Error Analysis

To better understand the behavior of the model, we performed a manual analysis of randomly sampled errors. We identified three major error types:

- **Ambiguous Knowledge:** The model struggled when multiple answers appeared plausible due to overlapping cultural concepts. For example, when asked about the founder of a specific Arab media outlet, the model confused the chief editor with the original founder.

- **Reasoning Gaps:** Some questions required multi step reasoning across history and religion, where the model failed to integrate knowledge.

- **Data Coverage Limitations:** Errors arise from missing representation of certain countries (e.g., Mauritania, Yemen) or underrepresented domains (e.g.,, folk traditions). This highlights the importance of broader cultural coverage in pre-training.

Overall, the errors suggest that while pretraining enriched the model with domain knowledge, deeper reasoning capabilities and broader cultural coverage remain key challenges.

5.4 Response Generation Quality

Our model successfully generates concise, accurate responses in the required format. Example model outputs demonstrate proper Arabic language usage and cultural sensitivity:

> **Input:**
> ما هو الطبق التقليدي الأشهر في المغرب العربي؟
> **Generated:** B
> **Gold:** B (الكسكس)

6 Conclusion

We presented a systematic approach to enhancing Arabic cultural understanding in language models through targeted pretraining and efficient fine-tuning. Our two-stage framework achieved competitive performance (74% development accuracy) while maintaining computational efficiency through LoRA adaptation.

Key contributions:

- A comprehensive cultural pretraining corpus spanning 10 Arab countries and more than 10 domains
- Demonstration that cultural pretraining significantly improves cultural QA performance
- An efficient training pipeline suitable for resource-constrained environments

Limitations:

- Limited coverage of dialectal variations across Arab regions
- Focus on factual knowledge may not capture implicit cultural understanding
- Performance gap between development and test sets suggests potential overfitting

Future Work: Future research directions include expanding corpus coverage to include dialectal content, investigating few-shot learning approaches for cultural adaptation, and developing more sophisticated evaluation metrics that capture cultural nuance beyond factual accuracy.

7 Acknowledgments

We thank the PalmX 2025 shared task organizers for providing this valuable evaluation framework. We also acknowledge the anonymous reviewers for their constructive feedback.

References

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025. PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic and Islamic Culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan AlRashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, and 1 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities. *arXiv preprint arXiv:2505.18383*.

Daniel Han, Michael Han, and Unsloth team. 2023. Unsloth. https://github.com/unslothai/unsloth.

Omer Nacar, Serry Taiseer Sibaee, Samar Ahmed, Safa Ben Atitallah, Adel Ammar, Yasser Alhabashi, Abdulrahman S Al-Batati, Arwa Alsehibani, Nour Qandos, Omar Elshehy, and 1 others. 2025. Towards inclusive arabic llms: A culturally aligned benchmark in arabic large language model evaluation. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 387–401.

Shravan Nayak, Kanishk Jain, Rabiul Awal, Siva Reddy, Sjoerd Van Steenkiste, Lisa Anne Hendricks, Aishwarya Agrawal, and 1 others. 2024. Benchmarking vision language models for cultural understanding. *arXiv preprint arXiv:2407.10920*.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2025. Survey of cultural awareness in language models: Text and beyond. *Computational Linguistics*, pages 1–96.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Kritsada Singhapoo, Akarachai Inthanil, and Attapon Pillai. 2025. Fine-tuning ai models with limited resources. In *2025 11th International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, pages 148–151. IEEE.

Jihoon Tack, Jack Lanchantin, Jane Yu, Andrew Cohen, Ilia Kulikov, Janice Lan, Shibo Hao, Yuandong Tian, Jason Weston, and Xian Li. 2025. Llm pretraining with continuous concepts. *arXiv preprint arXiv:2502.08524*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Dan Zhang, Tao Feng, Lilong Xue, Yuandong Wang, Yuxiao Dong, and Jie Tang. 2025. Parameter-efficient fine-tuning for foundation models. *arXiv preprint arXiv:2501.13787*.

AYA at PalmX 2025: Modeling Cultural and Islamic Knowledge in LLMs

Jannatul Tajrin[1*], **Bir Ballav Roy**[2], **Firoj Alam**[1]
[1]Qatar Computing Research Institute, Qatar
[2]BRAC University, Bangladesh
jannatultajrin33@gmail.com, bir.ballav.roy@g.bracu.ac.bd, fialam@hbku.edu.qa

Abstract

Culture fundamentally shapes human perception and reasoning, while religion—often embedded within cultural contexts—provides cohesive moral frameworks and a sense of community. The *PalmX 2025* shared task introduced two subtasks aimed at evaluating the capability of large language models (LLMs) to capture and represent culturally and Islamically grounded knowledge. In this paper, we present our participation in this shared task, leveraging parameter-efficient fine-tuning (PEFT) techniques in conjunction with targeted data augmentation strategies. We further conducted extensive zero-shot evaluations across a range of Arabic-centric and multilingual models to establish strong baselines and guide model selection. Our submitted system achieved competitive performance on the blind test sets, ranking 3^{rd} in Subtask 1 with an accuracy of 71.45% and 1^{st} in Subtask 2 with an accuracy of 84.22%.

1 Introduction

Culture is the shared system of meanings—values, norms, language, and rituals—that organizes how people perceive, decide, and relate (Hofstede, 2011). Religion, often a core strand of culture, provides moral frameworks, practices, and communities that guide conduct and purpose (Geertz, 2013). Attending to cultural and religious aspects improves communication, trust, and legitimacy, while reducing unintended harm and inequity across groups (Betancourt et al., 2003). Without culturally and religiously grounded priors, LLMs misinterpret idioms and taboos, amplify toxicity, and encode systematic biases, including documented anti-Muslim stereotypes (Gehman et al., 2020; Blodgett et al., 2020; Abid et al., 2021). Recent studies also show Western-leaning value biases and uneven cultural performance, demonstrating the need to encode diverse cultural and Islamic values in training data, safety policies, and evaluation suites (Li et al., 2024; Hasan et al., 2025).

To encode cultural, religious, and everyday knowledge, recent work has developed resources, methods, language-centric models, and benchmarks (Pawar et al., 2024). For Arabic, several LLMs have been pre-trained, including Jais (Sengupta et al., 2023), AceGPT (Huang et al., 2024), ALLaM (Bari et al., 2025), and Fanar (Team et al., 2025). While these models exhibit strong generative capabilities, many instruction-tuned variants rely heavily on synthetic or machine-translated data (e.g., Jais, AceGPT), which limits cultural knowledge and coverage. Moreover, most evaluations remain confined to general NLP and capability-oriented benchmarks (Abdelali et al., 2024; Mousi et al., 2025), with comparatively little attention to cultural and religious dimensions (Alwajih et al., 2025a). To advance the encoding of cultural and religious knowledge in Arabic-centric LLMs, the *PalmX 2025* shared task (Alwajih et al., 2025b) introduced a benchmark targeting Arabic cultural and islamic knowledge at both general and domain-specific levels, thereby enabling more inclusive and representative evaluations for the Arabic language and its diverse heritage. The shared task offered two subtasks. The annotated datasets for each subtask consists of human-validated multiple-choice question–answer pairs in MSA, ensuring both linguistic precision and cultural authenticity.

In this work, we benchmark multiple instruction-tuned LLMs across four configurations for both subtasks: (i) base, (ii) domain-specific fine-tuning, (iii) combined fine-tuning across subtasks, and (iv) data augmentation. Fine-tuning consistently improves performance on both subtasks: *Fanar-1-9B-Instruct* attains the higher accuracy on the cultural subtask (Subtask 1) under combined fine-tuning (80.8%), while *ALLaM-7B-Instruct* achieves the

[*] The contribution was made while the author was interning at the Qatar Computing Research Institute.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 830–836
November 8-9, 2025 ©2025 Association for Computational Linguistics

best accuracy on the Islamic subtask (Subtask 2) with augmented data (77.33%). Accordingly, we select the LoRA-based, combined fine-tuned Fanar model for Subtask1 and the ALLaM model with augmentation for Subtask 2 as our final systems. To summarize, our main contributions are:

- We present extensive baseline results for multiple LLMs under a zero-shot learning setup.
- Our proposed models achieved 3^{rd} place in Subtask 1 and 1^{st} place in Subtask 2.
- We show that paraphrase-based data augmentation yields notable performance gains for the islamic culture subtask.

2 Related Work

Recent advances in LLMs have demonstrated remarkable capabilities across a wide spectrum of natural language processing (NLP) tasks (Bubeck et al., 2023; Touvron et al., 2023; Abdelali et al., 2024; Dalvi et al., 2024). Beyond sheer model size, instruction tuning and preference optimization enhance both generalization and alignment, enabling models to follow user intent while delivering strong zero- and few-shot performance.

2.1 Cultural Knowledge

Recent work has begun to move beyond general Arabic capability benchmarks toward explicit evaluation of cultural competence. AraDiCE introduces a fine-grained dialect–culture suite spanning Gulf, Egypt, and Levantine, enabling targeted assessment of cultural awareness alongside dialectal understanding (Mousi et al., 2025). Country-specific evaluation is advancing as well: *SaudiCulture* focuses on regionally grounded cultural knowledge within Saudi Arabia (Ayash et al., 2025). Another recent effort proposed a framework, which highlights the significance of benchmarking LLMs with culturally embraced data, underlining the performance disparity between high and low resource language (Alam et al., 2025; Hasan et al., 2025). These efforts complement broader Arabic benchmarks such as LAraBench, which established multi-task capability evaluations but did not directly target cultural facets (Abdelali et al., 2024).

2.2 Islamic Knowledge

In contrast to the breadth of general cultural evaluation, Islamic/religious benchmarking remains limited in scale and linguistic coverage. *QUQA* evaluates GPT-4 on Classical-Arabic Qur'anic QA and reports modest exact-match and F1 scores, revealing limits even for state-of-the-art models in scripture-centric settings (Alnefaie et al., 2023). A cross-lingual Qur'anic QA effort expands a small Arabic set to 1,895 Arabic–English pairs and assesses pre-trained LMs/LLMs mainly with retrieval metrics (MAP@10, MRR, Recall@10), offering a first but narrow bilingual baseline (Oshallah et al., 2025). Retrieval-augmented studies over Qur'anic summaries examine faithfulness and citation via human ratings for open-source LLMs, but remain English-only and task-specific (Khalila et al., 2025). Multimodal cultural VQA benchmarks include religious practices and iconography across many languages; however, they target vision–language models and do not provide text-only, source-grounded Islamic QA suitable for doctrinal assessment (Vayani et al., 2025).

On the resource side, *Hajj-FQA* offers a human-annotated QA set over Hajj fatwas (Aleid and Azmi, 2025); *Fatwaset* compiles a large Arabic fatwa corpus with rich metadata for downstream NLP (Alyemny et al., 2023); and Qur'anic QA resources—such as the Qur'anic Reading Comprehension Dataset (QRCD) and subsequent retrieval/QA studies—provide task-specific testbeds while exposing issues of hallucination and domain brittleness (Basem et al., 2025). Collectively, these works lay important groundwork for Islamic-knowledge evaluation in Arabic; nevertheless, coverage remains narrow (few languages beyond Arabic/English), datasets are modest, and critical scholarly dimensions—madhhab/fiqh context, *hadith* authenticity, *tafsīr* grounding, awareness of abrogation (*naskh*), and dialectal/diacritic variation—are largely unencoded, leaving a clear gap relative to the methodological rigor now common in broader multicultural evaluation.

3 Tasks and Dataset

3.1 Tasks

The PalmX 2025 Shared Task evaluates Arabic General culture and Islamic knowledge through multiple-choice question answering in Modern Standard Arabic. It consists of two subtasks:

- **Subtask 1 – General Culture Evaluation:** This subtask evaluates the ability of LLMs to comprehend and reason about diverse aspects of Arabic general culture, including traditional customs, social etiquette, cuisine, historical events, notable figures, geography, arts, and dialectal expressions across different Arab countries. The focus is on assessing models' capacity to apply broad cultural knowledge that is relevant across the Arab world.

- **Subtask 2 – General Islamic Evaluation:** This subtask measures models' understanding of core elements of Islamic culture, which forms a foundational component of many Arabic societies. It covers topics such as Islamic rituals and practices (e.g., prayer, fasting), Quranic knowledge, Hadith literature, major historical developments in Islam, and religious holidays. Models are evaluated on multiple-choice questions designed to test both religious literacy and contextual sensitivity, ensuring they can handle culturally and theologically significant content with accuracy and respect.

3.2 Dataset

We used the dataset released as a part of PalmX 2025 Shared Task. For both subtasks the datasets has been formulated as MCQ format.

Data Augmentation. We employed paraphrase-based data augmentation to increase the diversity and robustness of the questions in the training data. In this approach, original questions were reworded into semantically equivalent variants while strictly preserving their intended meaning and correct answers. The resulting augmented dataset introduced controlled variations in phrasing, complexity, and syntactic structure, thereby encouraging better generalization. We used the GPT-4.1 model to paraphrase the questions. Listing 1 shows the exact prompt we used.

In Table 1, we report the detail distribution of the dataset. As reported in the Table, for both subtasks we applied data augmentation to increase training set size. As for the development and test set we have used same dataset released as a part of the shared task. Test set in the table refers to the blind test set. Note that in our initial set of experiments we have used dev set as a test set to evaluate models' performance.

Listing 1: Prompt for paraphrase based data augmentation.

```
system_prompt = (
    "You are a high-quality data augmentation
    assistant for Arabic multiple-choice
    question answering. "
    "Your job is to create adversarial variants
    of questions: rephrase or make the question
    more challenging "
    "or tricky, but do not alter its meaning or
    change which answer is correct. "
    "The answer options and the correct answer
    must remain valid for the new question."
    )
user_prompt = (
        "Below is an Arabic multiple-choice
        question with options and the correct
        answer indicated. "
        "Rewrite the question to make it
        slightly more challenging or confusing
        for test-takers "
        "(e.g., use more complex language, add
        subtle ambiguity, or require deeper
        understanding), "
        "but do not change its intended meaning
        or the correct answer. "
        "Return your answer as a JSON object
        with the new question, all original
        options, and answer letter
        preserved.\n\n"
        f"Question: {data['question']}\n"
        f"A: {data['A']}\n"
        f"B: {data['B']}\n"
        f"C: {data['C']}\n"
        f"D: {data['D']}\n"
        f"Answer: {data['answer']}"
    )
```

Subtasks	Train	Dev	Test
Culture	2,000	500	2,000
Culture + Islamic	2,600	500	2,000
Culture + Aug	4,000	500	2,000
Islam	600	300	1,000
Islam + Aug	1,200	300	1,000

Table 1: Dataset statistics for PalmX 2025 subtasks. Aug refers to data augmentation.

4 Experiments

4.1 Models

We have selected several instruction-tuned LLMs for the zero-shot evaluation and fine-tuning models on two subtasks – *General Cultural* and *Islamic Cultural* – under four configurations: base, fine-tuned, combined fine-tuning (culture + Islamic), and augmented data. The models considered included `Qwen2.5-7B-Instruct` (Wang et al., 2024), `Jais-13B-Chat` (Sengupta et al., 2023), `Miraj Mini`,[1] `Llama-3.1-8B-Instruct` (Touvron et al., 2023), `NileChat-3B` (Mekki et al., 2025), `ALLaM-7B-Instruct` (Bari et al., 2025), Gemma-7b-it[2] and `Fanar-7B-Instruct` (Team et al., 2025).

[1] https://huggingface.co/arcee-ai/Meraj-Mini

[2] https://huggingface.co/google/gemma-7b-it

4.2 Training

We fine-tuned the models using the LoRA approach. The LoRA configuration used a rank (r) of 16, an alpha value of 32, a base learning rate of 2e-4 and a dropout rate of 0.05, a maximum sequence length of 512 tokens, trained for three epochs, targeting the query and value projection layers of the transformer architecture. LoRA adapters were loaded from a prior checkpoint and the implementation of *attention* was set to *'eager'* for compatibility. The tokenizer for the base model was used, with the padding token aligned with the end-of-sequence token. The evaluation was performed with a batch size of 4 to accommodate the memory requirements of the 9B parameter model. The prompts were formatted for multiple choice answer with predefined choice prefixes (A, B, C, D).

Evaluation We evaluated models using accuracy, calculated as the percentage of correctly answered questions. For model training and internal evaluation, we were limited to the development dataset. Final evaluation and ranking were carried out by the organizers on the blind test set.

5 Results

The evaluation results across the PalmX subtasks demonstrate notable improvements through fine-tuning and data combination.

Cultural Subtask. In Table 2, we report the performance of cultural evaluation on the development set. For the PalmX General Cultural subtask, the ALLaM-7B-Instruct model improved from a base accuracy of 63.8 to 75.8 after fine-tuning, maintaining the same performance when combined with the Islamic dataset. Fanar-1-9B-Instruct outperformed ALLaM on this subtask, achieving 72.4 at base and improving to 80.2 after fine-tuning, with a slight increase to 80.8 using the combined data.

Islamic Subtask. In Table 3, we report the performance of Islamic evaluation on the development set. In the PalmX Islamic Culture subtask, ALLaM-7B showed a base accuracy of 72.7, which improved to 76.33 after fine-tuning and further to 77.33 with additional Islamic data augmentation. On the final hidden test set, Fanar-1-9B achieved an accuracy of 71.45 on the General Culture evaluation, while ALLaM-7B attained 84.22 on the General Islamic evaluation, indicating strong performance in their respective domains.

Error analysis. Figure 1, in Appendix, presents the confusion matrix for Subtask 1 on the hidden test set. The raw confusion matrix shows the proportion of correct and incorrect predictions per true label. Off-diagonal entries reveal misclassification patterns, with slightly higher confusion between classes **A** and **B** as well as **C** and **D**. Overall, the model demonstrates balanced accuracy across categories, with no single class dominating the error distribution.

Model	Base	Train Set	Comb.	Aug.
Gemma-7B-it	49.6	67.6	39.6	49.6
ALLaM-7B	63.8	75.8	75.8	70.6
Llama3.1-8B	66.6	66.6	66.6	66.6
Qwen2.5-7B	69.2	69.2	69.2	69.2
NileChat-3B	70.0	76.0	72.8	70.0
Fanar-1-9B	72.4	80.2	**80.8**	79.2

Table 2: Results on development set with a comparison of different models on PalmX *General Cultural subtask*. FT: Fine-tuned only on the PalmX training set, Comb.: Combined (Culture + Islamic), Aug.: Cultural + Augmented.

Model	Base	Train Set	Comb.	Aug.
Gemma-7B-it	40.0	40.0	25.3	25.3
Qwen2.5-7B	57.3	57.3	57.3	57.3
NileChat-3B	64.0	64.0	64.0	63.7
Fanar-1-9B	67.0	66.0	70.3	67.0
ALLaM-7B	72.7	76.3	76.3	**77.3**

Table 3: Performance comparison of different models on PalmX *Islamic Cultural subtask*. Comb.: Culture + Islamic, Aug.: Islamic + Augmented.

Subtasks	Model	Acc
Culture	Fanar-1-9B	71.5
Islamic	ALLaM-7B	84.2

Table 4: Performance on final hidden test set.

Figure 2, in Appendix, illustrates the confusion matrix for Subtask 2 on the hidden test set. The model correctly predicted **127**, **483**, **179**, and **54** instances for classes **A**, **B**, **C**, and **D**, respectively. The largest proportion of correct predictions occurred for class B, with relatively low misclassification rates. Notable confusion patterns include A → B (32 instances) and D → B (10 instances). While diagonal dominance is evident, indicating that the model captures the underlying class distinctions well, performance on class D is comparatively lower, suggesting a need for more targeted learning on that category. Overall, the results reflect strong performance, with class B exhibiting the highest classification accuracy in the Islamic

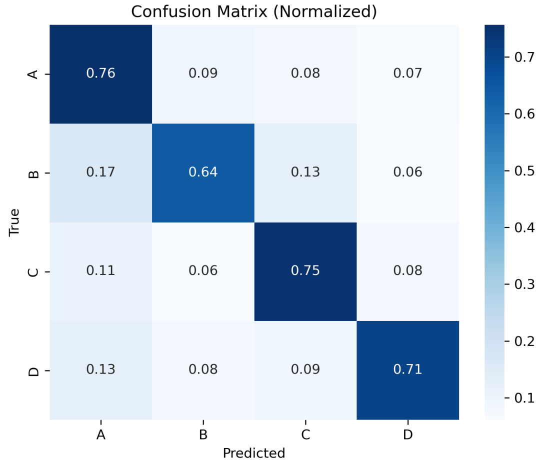


Figure 1: Confusion matrices for Subtask 1 (General Cultural Evaluation) on the hidden test set.

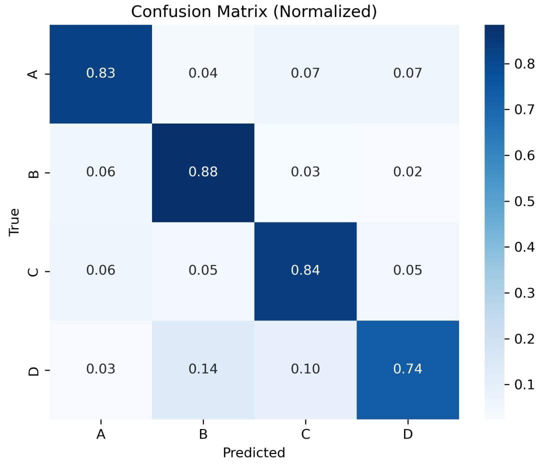


Figure 2: Confusion matrices for Subtask 2 (Islamic Cultural Evaluation) on the hidden test set.

cultural knowledge domain.

6 Conclusions and Future Work

In this paper, we presented our system developed for the Palmx 2025 Shared Task on MSA multiple-choice question answering in the Cultural and Islamic Evaluation. Our approach focused on fine-tuning state-of-the-art large language models, specifically ALLaM-7B-Instruct and Fanar-1-9B-Instruct, with data augmentation on domain-specific datasets, leveraging combined cultural and Islamic data to enhance performance. The Fanar-1-9B-Instruct model achieved the highest accuracy on the General Cultural subtask with 80.8 after fine-tuning and data combination, while ALLaM-7B-Instruct showed strong results in the Islamic subtask, reaching 77.33 accuracy with augmented data. On the final hidden test set, Fanar-1-9B-Instruct scored 71.45 on the General Culture evaluation, and ALLaM-7B-Instruct achieved 84.22 accuracy on the General Islamic evaluation. These results demonstrate the effectiveness of fine-tuning and data augmentation strategies in improving performance across different subtasks.

7 Limitations

While our experiments provide valuable insights into cultural and Islamic evaluation tasks, several limitations remain. Despite dataset variations (combined Islamic + General Cultural data and augmentation), some models such as *google/gemma-7b-it* and *Qwen2.5-7B-Instruct* showed nearly identical accuracy across settings, indicating limited sensitivity to data scale or diversity. We also observed accuracy decreases when training on combined datasets. However, performance overall across training setups is better than the base (unfine-tuned) model. These findings highlight the need for deeper error analysis, improved fine-tuning methods, and more robust data integration to better adapt language models for nuanced cultural understanding.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Samir Abdaljalil, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, Nizi Nazar, Youssef Elshahawy, Ahmed Ali, Nadir Durrani, Natasa Milic-Frayling, and Firoj Alam. 2024. LAraBench: Benchmarking Arabic AI with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–520, St. Julian's, Malta. Association for Computational Linguistics.

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 298–306.

F. Alam, Md Asif Hasan, S. R. Laskar, M. Kutlu, Kareem Darwish, and S. A. Chowdhury. 2025. NativQA framework: Enabling llms with native, local, and everyday knowledge. *arXiv preprint arXiv:2504.05995*.

Hayfa A. Aleid and Aqil M. Azmi. 2025. Hajj-FQA: A benchmark arabic dataset for developing question-answering systems on hajj fatwas. *Journal of King Saud University - Computer and Information Sciences*, 37:Article 135.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Is GPT-4 a good islamic expert for answering Quran questions? In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 124–133, Taipei City, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Fakhraddin Alwajih, Abdellah El Mekki, Samar Mohamed Magdy, AbdelRahim A. Elmadany, Omer Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, Abdulfattah Mohammed Yahya, Rahaf Alhamouri, Hamzah A. Alsayadi, Hiba Zayed, Sara Shatnawi, Serry Sibaee, Yasir Ech-chammakhy, Walid Al-Dhabyani, Marwa Mohamed Ali, Imen Jarraya, and 25 others. 2025a. Palm: A culturally inclusive and linguistically diverse dataset for Arabic LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32871–32894, Vienna, Austria. Association for Computational Linguistics.

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025b. PalmX 2025: The first shared task on benchmarking llms on arabic and islamic culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Ohoud Alyemny, Hend Al-Khalifa, and Abdulrahman Mirza. 2023. A data-driven exploration of a new islamic fatwas dataset for arabic nlp tasks. *Data*, 8(10):155.

Lama Ayash, Hassan Alhuzali, Ashwag Alasmari, and Sultan Aloufi. 2025. Saudiculture: A benchmark for evaluating large language models' cultural competence within saudi arabia. *Journal of King Saud University Computer and Information Sciences*, 37(6):123.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham Abdullah Alyahya, Sultan AlRashed, Faisal Abdulrahman Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Saad Amin Hassan, Dr. Majed Alrubaian, Ali Alammari, Zaki Alawami, and 7 others. 2025. ALLam: Large language models for arabic and english. In *The Thirteenth International Conference on Learning Representations*.

Mohamed Basem, Islam Oshallah, Ali Hamdi, and Ammar Mohammed. 2025. Few-shot prompting for extractive quranic qa with instruction-tuned llms. *arXiv preprint arXiv:2508.06103*.

Joseph R Betancourt, Alexander R Green, J Emilio Carrillo, and Owusu Ananeh-Firempong 2nd. 2003. Defining cultural competence: a practical framework for addressing racial/ethnic disparities in health and health care. *Public health reports*, 118(4):293.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. Technical report, Microsoft Research.

Fahim Dalvi, Maram Hasanain, Sabri Boughorbel, Basel Mousi, Samir Abdaljalil, Nizi Nazar, Ahmed Abdelali, Shamur Absar Chowdhury, Hamdy Mubarak, Ahmed Ali, Majd Hawasly, Nadir Durrani, and Firoj Alam. 2024. LLMeBench: A flexible framework for accelerating llms benchmarking. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, Malta. Association for Computational Linguistics.

Clifford Geertz. 2013. Religion as a cultural system. In *Anthropological approaches to the study of religion*, pages 1–46. Routledge.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Md. Arid Hasan, Maram Hasanain, Fatema Ahmad, Sahinur Rahman Laskar, Sunaya Upadhyay, Vrunda N Sukhadia, Mucahid Kutlu, Shammur Absar Chowdhury, and Firoj Alam. 2025. NativQA: Multilingual culturally-aligned natural query for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14886–14909, Vienna, Austria. Association for Computational Linguistics.

Geert Hofstede. 2011. Dimensionalizing cultures: The hofstede model in context. *Online readings in psychology and culture*, 2(1):8.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Song Dingjie, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 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 (Volume 1: Long Papers)*, pages 8139–8163, Mexico City, Mexico. Association for Computational Linguistics.

Zahra Khalila, Arbi Haza Nasution, Winda Monika, Aytug Onan, Yohei Murakami, Yasir Bin Ismail Radi, and Noor Mohammad Osmani. 2025. Investigating retrieval-augmented generation in quranic studies: A study of 13 open-source large language models. *arXiv preprint arXiv:2503.16581*.

Cheng Li, Damien Teney, Linyi Yang, Qingsong Wen, Xing Xie, and Jindong Wang. 2024. CulturePark: Boosting cross-cultural understanding in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 65183–65216.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. NileChat: Towards linguistically diverse and culturally aware llms for local communities. *arXiv preprint arXiv:2505.18383*.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2025. AraDiCE: Benchmarks for dialectal and cultural capabilities in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4186–4218, Abu Dhabi, UAE. Association for Computational Linguistics.

Islam Oshallah, Mohamed Basem, Ali Hamdi, and Ammar Mohammed. 2025. Cross-language approach for quranic qa. *arXiv preprint arXiv:2501.17449*.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2024. Survey of cultural awareness in language models: Text and beyond. *arXiv preprint arXiv:2411.00860*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, and 1 others. 2023. Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Fanar Team, 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, and 23 others. 2025. Fanar: An arabic-centric multimodal generative ai platform.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashmal Vayani, Dinura Dissanayake, Hasindri Watawana, Noor Ahsan, Nevasini Sasikumar, Omkar Thawakar, Henok Biadglign Ademtew, Yahya Hmaiti, Amandeep Kumar, Kartik Kukreja, and 1 others. 2025. All languages matter: Evaluating lmms on culturally diverse 100 languages. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19565–19575.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Cultura-Arabica: Probing and Enhancing Arabic Cultural Awareness in Large Language Models via LoRA[*]

Pulkit Chatwal and **Santosh Kumar Mishra**
Rajiv Gandhi Institute of Petroleum Technology, Jais, India
pulkitchatwal@gmail.com and satosh.mishra@rgipt.ac.in

Abstract

Large Language Models (LLMs) have demonstrated impressive multilingual capabilities; however, their reasoning often reflects English-centric perspectives, which can limit accuracy in culture-specific contexts. Arabic, with its diverse dialects, rich historical heritage, and complex socio-cultural norms, presents a particularly challenging setting for such evaluation. To address this gap, we participated in the PalmX 2025 shared task, which benchmarks cultural reasoning in Arabic through multiple-choice questions covering traditions, social norms, history, geography, arts, and dialectal expressions. By applying parameter-efficient adaptation and culturally informed prompt formatting, we aligned model outputs with both linguistic correctness and cultural relevance. Our approach achieved an accuracy of **71.65%**, securing **second place** overall and closely matching the top system. These results demonstrate that targeted adaptation can significantly enhance cultural reasoning in LLMs, paving the way for more culturally aware Artificial Intelligence.

1 Introduction

Large Language Models (LLMs) have transformed natural language processing, excelling in multilingual understanding, reasoning, and text generation (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). Yet, their reasoning often reflects predominantly English-centric worldviews (Bang et al., 2023; Piqueras and Søgaard, 2022), leading to gaps in interpreting culture-specific knowledge, norms, and perspectives. Cultural reasoning—integrating linguistic comprehension with contextual understanding of traditions, values, and social practices—is essential for fair, contextually appropriate AI systems (Tao et al., 2024).

Arabic, with its diverse dialects, historical depth, and socio-cultural richness, is a particularly challenging testbed. Despite the growth of Arabic NLP resources, most models remain optimized for syntactic and semantic accuracy rather than capturing the implicit socio-cultural knowledge needed to interpret idioms, customs, and worldview-specific references. As Marcus and Davis emphasize, LLMs are powerful pattern recognizers but lack genuine understanding and grounded reasoning, often reproducing correlations without true comprehension (Marcus and Davis, 2019). Recent work also shows that "models tend to exhibit Western bias even when prompted in non-English languages like Arabic" (Naous et al., 2023), underscoring persistent cultural blind spots.

Addressing this challenge requires moving beyond language correctness toward genuine cultural alignment—where models reason in ways consistent with the target community's norms and context. This work examines whether parameter-efficient adaptation can improve the cultural reasoning capabilities of Arabic LLMs, bridging the gap between linguistic competence and culturally grounded intelligence.

2 Related Work

Arabic Natural Language Processing (NLP) has advanced notably in recent years, driven by transformer-based architectures, culturally aligned datasets, and resource-efficient adaptation methods.

AraBERT (Antoun et al., 2020) pioneered Arabic-specific BERT pre-training, achieving state-of-the-art results in sentiment analysis, named entity recognition, and question answering. ARBERT and MARBERT (Abdul-Mageed et al., 2020) extended this to Modern Standard Arabic (MSA) and dialects, accompanied by ARLUE, a benchmark for multi-dialectal understanding. These works underscore the value of Arabic-specific pre-training.

[*] The final fine-tuned model is available at https://huggingface.co/Pulkit-28/PalmQA-3B-Arabic.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 837–842
November 8-9, 2025 ©2025 Association for Computational Linguistics

Culturally grounded datasets have emerged to address linguistic and cultural biases. CIDAR (Alyafeai et al., 2024) is the first open Arabic instruction-tuning dataset curated for cultural relevance, improving alignment of large language models (LLMs) with Arabic norms. Other domain-specific benchmarks include AraSTEM (Mustapha et al., 2024) for STEM knowledge and AlGhafa (Almazrouei et al., 2023) for diverse Arabic MCQs. Beyond Arabic, the *Survey of Cultural Awareness in Language Models* (Pawar et al., 2025) reviews methods for integrating cultural sensitivity into text and multimodal LLMs, with discussion of datasets, benchmarking, and ethics.

Resource-efficient fine-tuning has also gained traction. Low-Rank Adaptation (LoRA) (Hu et al., 2022) reduces trainable parameters while maintaining performance, and Quantized Low-Rank Adaptation (QLoRA)-based adaptation for Arabic (Aryan, 2024) achieves high-quality results with minimal hardware. Parameter-efficient methods have also been applied to dialect identification (Radhakrishnan et al., 2023) with competitive accuracy.

Large-scale Arabic foundation models like Jais and Jais-chat (Sengupta et al., 2023) set records in Arabic reasoning tasks, while LAraBench (Abdelali et al., 2023) offers a comprehensive benchmarking suite for Arabic NLP and speech, revealing gaps between general-purpose and specialized Arabic models. Beyond Arabic, *Beyond English-Centric LLMs* (Zhong et al., 2024) shows multilingual models may rely on multiple latent languages, stressing the need to study internal representation dynamics for better cultural adaptation.

In summary, advances in Arabic NLP arise from the synergy of specialized pre-training, culturally relevant datasets, efficient fine-tuning, and robust benchmarking—together enhancing accuracy, cultural sensitivity, and efficiency in Arabic-focused LLMs.

3 Problem Statement

We participated in the PalmX 2025 shared task (Alwajih et al., 2025), which evaluates large language models (LLMs) on their ability to comprehend and reason about *Arabic general culture*—including traditions, social norms, history, geography, arts, and dialectal variations. Formally, let $\mathcal{Q} = \{q_1, \dots, q_n\}$ be a set of culturally grounded questions in Modern Standard Arabic, each with candidate answers $\mathcal{A}_i$, where exactly one a_i^* is correct. An LLM, modeled as $f_\theta : \mathcal{Q} \to \mathcal{A}$, aims to maximize:

$$\frac{1}{n} \sum_{i=1}^{n} \nVdash\{\hat{a}_i = a_i^*\}.$$

Unlike traditional benchmarks that focus on $P(\hat{a}_i = a_i^* \mid \text{linguistic knowledge})$, this task emphasizes $P(\hat{a}_i = a_i^* \mid \text{linguistic knowledge, cultural knowledge})$, ensuring models are both linguistically accurate and culturally grounded.

4 Dataset

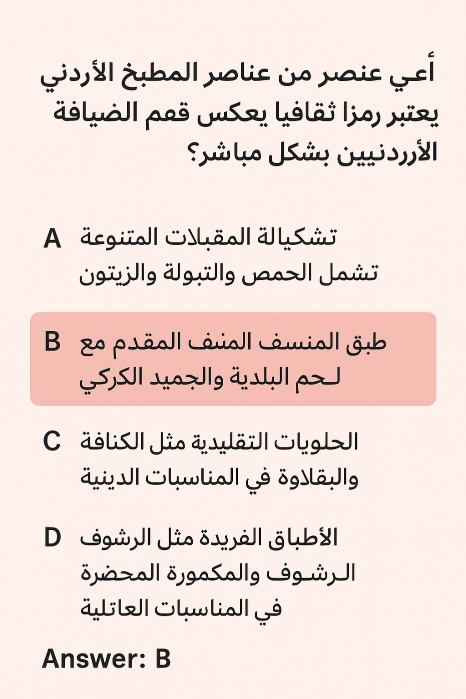
The dataset provided for the PalmX 2025 shared task (Alwajih et al., 2025) was specifically curated to evaluate cultural reasoning capabilities in Arabic LLMs. It consists of three partitions, each balanced across domains such as traditions, social norms, history, geography, arts, and dialectal expressions from diverse Arab countries. The statistics of the dataset are summarized in Table 1, and an example from the training set is shown in Figure 1.

Partition	Number of MCQs
Training set	2,000
Development set	500
Blind test set	2,000

Table 1: Summary statistics of the dataset provided for the PalmX 2025 shared task

5 Methodology

This section delineates the modeling framework employed to adapt a large Arabic language model for the PalmX 2025 cultural reasoning task. Recognizing that the task entails selecting the appropriate option from multiple culturally grounded choices, we formulate it as a causal language modeling problem augmented with structured prompts. This approach not only facilitates the model's acquisition of reasoning patterns encompassing both factual and cultural knowledge but also leverages the inherent generative capabilities of language models to handle nuanced, context-dependent queries effectively.



أعـي عنصر من عناصر المطبخ الأردني
يعتبر رمزا ثقافيا يعكس قمم الضيافة
الأرردنيين بشكل مباشر؟

A تشكيالة المقبلات المتنوعة
تشمل الحمص والتبولة والزيتون

B طبق المنسف المنف المقدم مع
لـحم البلدية والجميد الكركي

C الحلويات التقليدية مثل الكنافة
والبقلاوة في المناسبات الدينية

D الأطباق الفريدة مثل الرشوف
الـرشـوف والمكمورة المحضرة
في المناسبات العاتلية

Answer: B

Figure 1: Sample culturally grounded MCQ from the training set.

5.1 Base Model

Our framework is built upon the `NileChat-3B` checkpoint (Mekki et al., 2025), a 3B-parameter decoder-only transformer specifically optimized for Arabic dialogue and general-purpose text generation. This model was selected due to its robust pre-training on a diverse corpus of Arabic text, which includes dialectal variations and cultural contexts, making it particularly suitable for tasks requiring deep linguistic and sociocultural understanding. The architecture adheres to an autoregressive GPT-style design, comprising 24 stacked multi-head self-attention layers interspersed with feed-forward blocks, all geared toward efficient left-to-right token prediction. The tokenizer, derived from the same checkpoint, utilizes byte-pair encoding (BPE) with a vocabulary size of 50,000 tokens to accommodate both Arabic and non-Arabic scripts, with the end-of-sequence (EOS) token repurposed as the padding token to ensure seamless compatibility with causal modeling paradigms.

5.2 Fine-Tuning Strategy

For efficient adaptation, we leverage Low-Rank Adaptation (LoRA) (Hu et al., 2022), a parameter-efficient fine-tuning technique that introduces trainable low-rank decomposition matrices into the transformer's projection layers while keeping the original weights frozen. This method allows us to fine-tune fewer than 1% of the total parameters, achieving an optimal trade-off between computational overhead and expressive capacity, which is especially beneficial for resource-constrained environments and multilingual models where full fine-tuning could lead to catastrophic forgetting of pre-trained knowledge. The specific LoRA configuration adopted in this study is as follows:

- Rank (r): 16
- Scaling factor (α): 32
- Target modules: `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`
- Dropout rate: 0.05
- Bias: none

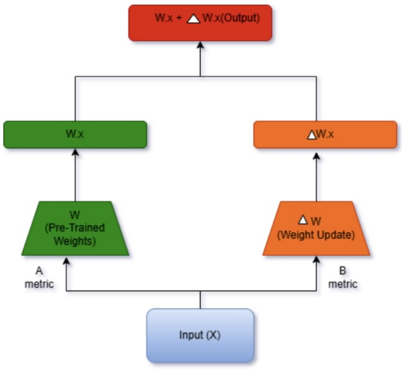


Figure 2: Schematic illustration of the Low-Rank Adaptation (LoRA) mechanism integrated into the fine-tuning process.

5.3 Prompt Formatting

To optimize the model's performance on the multiple-choice cultural reasoning task, each dataset instance is converted into a carefully designed prompt structure. This includes the question stem, four labeled options (A through D), and a clear instruction to generate only the letter of the correct choice. An illustrative prompt template is as follows:

```
Question: [Question text]
Options:
A) [Option A]
B) [Option B]
C) [Option C]
D) [Option D]
Output only the correct letter:
```

This structured format reduces output variability during inference, promotes focused discriminative reasoning by the model, and ensures tight alignment between the training objective and prevalent evaluation paradigms in cultural reasoning, such as zero-shot or few-shot settings.

5.4 Training Procedure

The fine-tuning process is executed via the `Transformers` library's `Trainer` API, incorporating mixed-precision training in `bfloat16` to enhance computational efficiency and reduce memory footprint. We utilize a per-device batch size of 2 on a single NVIDIA A100 GPU, augmented by gradient accumulation across 4 steps, resulting in an effective batch size of 8. A fixed learning rate of 2×10^{-4} is applied without scheduling, with training spanning three epochs to balance convergence and overfitting prevention. The `DataCollatorForLanguageModeling` is configured with `mlm=False` to uphold the causal autoregressive training objective, ensuring that the model learns to generate responses conditioned on the full prompt context. Throughout training, we monitor validation loss to confirm generalization to unseen cultural reasoning examples.

5.5 Adapter Merging and Deployment

Upon completion of fine-tuning, the LoRA adapters are integrated into the base model weights through the `merge_and_unload()` procedure, yielding a consolidated checkpoint devoid of external dependencies and maintaining the original model's inference speed. This merging step is crucial for production environments, as it eliminates the need for additional adapter loading during deployment. The resultant model, designated as `NileChat-3B-Arabic-QA-Merged-v2`, is primed for seamless inference and deployment in practical applications, such as interactive cultural education tools or multilingual question-answering systems.

6 Results

6.1 Evaluation

The official metric for the PalmX 2025 shared task was *accuracy*, measuring the proportion of correct predictions across all test questions. Given a test set of N questions, accuracy is calculated as:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \ (\hat{y}_i = y_i)}{N} \times 100\%, \qquad (1)$$

where $\hat{y}_i$ denotes the predicted answer for question i, y_i represents the gold standard label, and $\ (\cdot)$ is the truth indicator returning 1 if the argument is true and 0 otherwise. This metric equally weights all questions, ensuring that performance reflects general reasoning capabilities rather than domain-specific biases.

6.2 Leaderboard Performance

Our system obtained an overall accuracy of **71.65%**, securing the **second rank** among all participating teams. This performance demonstrates that our parameter-efficient LoRA fine-tuning method can effectively adapt a large Arabic LLM to culturally grounded multiple-choice reasoning with limited task-specific data.

Rank	Team	Score (%)
1	HAI research group	72.15
2	**Our Result**	**71.65**
3	AYA_Team	71.45
4	Phoenix	71.35
5	CultranAI	70.50
6	ISL-NLP	67.60
7	Rafiul Biswas	67.55
8	Hamyaria	65.90
9	Star	64.05

Table 2: Leaderboard results from the PalmX 2025 shared task.

6.3 Discussion

The narrow margin between the top three teams—less than one percentage point—indicates that small architectural or fine-tuning choices can substantially influence outcomes in culturally nuanced reasoning tasks. Our approach's ability to match and even surpass larger-scale fine-tuning efforts highlights the efficiency of targeted LoRA adaptation for Arabic cultural QA, while suggesting broader implications for resource-efficient multilingual NLP.

7 Future Work

Promising directions for extending this work include adapting the proposed framework to other low-resource languages, thereby assessing its efficacy in cross-lingual cultural reasoning tasks. Furthermore, integrating multimodal capabilities—such as fine-tuning on Visual Question Answering (VQA) datasets enriched with culturally pertinent images—could substantially improve model performance by synergistically combining visual and textual cues for more nuanced cultural understanding.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Absar Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, et al. 2023. Larabench: Benchmarking arabic ai with large language models. *arXiv preprint arXiv:2305.14982*.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Ebtesam Almazrouei, Ruxandra Cojocaru, Michele Baldo, Quentin Malartic, Hamza Alobeidli, Daniele Mazzotta, Guilherme Penedo, Giulia Campesan, Mugariya Farooq, Maitha Alhammadi, et al. 2023. Alghafa evaluation benchmark for arabic language models. In *Proceedings of ArabicNLP 2023*, pages 244–275.

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025. PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic and Islamic Culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Zaid Alyafeai, Khalid Almubarak, Ahmed Ashraf, Deema Alnuhait, Saied Alshahrani, Gubran AQ Abdulrahman, Gamil Ahmed, Qais Gawah, Zead Saleh, Mustafa Ghaleb, et al. 2024. Cidar: Culturally relevant instruction dataset for arabic. *arXiv preprint arXiv:2402.03177*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Prakash Aryan. 2024. Resource-aware arabic llm creation: Model adaptation, integration, and multi-domain testing. In *International Conference on Advanced Network Technologies and Intelligent Computing*, pages 415–434. Springer.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Gary Marcus and Ernest Davis. 2019. *Rebooting AI: Building artificial intelligence we can trust*. Vintage.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities.

Ahmad Mustapha, Hadi Al-Khansa, Hadi Al-Mubasher, Aya Mourad, Ranam Hamoud, Hasan El-Husseini, Marwah Al-Sakkaf, and Mariette Awad. 2024. Arastem: A native arabic multiple choice question benchmark for evaluating llms knowledge in stem subjects. *arXiv preprint arXiv:2501.00559*.

Tarek Naous, Michael J Ryan, Alan Ritter, and Wei Xu. 2023. Having beer after prayer? measuring cultural bias in large language models. *arXiv preprint arXiv:2305.14456*.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2025. Survey of cultural awareness in language models: Text and beyond. *Computational Linguistics*, pages 1–96.

Laura Cabello Piqueras and Anders Søgaard. 2022. Are pretrained multilingual models equally fair across languages? *arXiv preprint arXiv:2210.05457*.

Srijith Radhakrishnan, Chao-Han Huck Yang, Sumeer Ahmad Khan, Narsis A Kiani, David Gomez-Cabrero, and Jesper N Tegner. 2023. A parameter-efficient learning approach to arabic dialect identification with pre-trained general-purpose speech model. *arXiv preprint arXiv:2305.11244*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, et al. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Yan Tao, Olga Viberg, Ryan S Baker, and René F Kizilcec. 2024. Cultural bias and cultural alignment of large language models. *PNAS nexus*, 3(9):pgae346.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Chengzhi Zhong, Fei Cheng, Qianying Liu, Junfeng Jiang, Zhen Wan, Chenhui Chu, Yugo Murawaki, and Sadao Kurohashi. 2024. Beyond english-centric llms: What language do multilingual language models think in? *arXiv preprint arXiv:2408.10811*.

Phoenix at Palmx: Exploring Data Augmentation for Arabic Cultural Question Answering

Houdaifa Atou$^{\lambda*}$**, Issam Ait Yahia**$^{\lambda*}$**, Ismail Berrada**$^{\lambda}$**,**
$^{\lambda}$College of Computing
Mohammed VI Polytechnic University, Ben Guerir, Morocco
{houdaifa.atou, issam.aityahia, ismail.berrada}@um6p.ma

Abstract

Large Language Models (LLMs) have become central to natural language processing, but their performance in low-resource cultural domains remains limited, mainly due to the dominance of English data in training. This limitation is especially evident in open small models. Evaluating and improving LLMs' performance in Arabic culture is therefore necessary. This paper presents ***Phoenix*** and ***PhoenixIs***, two models fine-tuned for the ***Palmx 2025*** general culture and Islamic culture subtasks. Phoenix uses the *Palmx-GC* and *Palmx-IC* datasets as seed data and applies diverse data augmentation strategies to construct an enriched fine-tuning dataset. *Phoenix* achieves an accuracy of 71.35% on the general culture subtask, while *PhoenixIs* reaches 83.82% on the Islamic culture subtask.

1 Introduction

Culture refers to the shared knowledge, beliefs, values, practices, and traditions that shape how a community understands and interacts with the world. Although Large Language Models (LLMs) have achieved strong performance across a wide range of natural language processing tasks, they have been shown to exhibit cultural bias toward Western culture (Cecilia Liu et al., 2024; Navigli et al., 2023; Cao et al., 2023). Such bias arises from the dominance of English data in their pre-training and post-training corpora. This limitation may affect their ability to adapt and generate culturally appropriate responses for diverse communities. To tackle this bias, efforts have been made to align LLMs with different cultures (Joshi et al., 2025; Mekki et al., 2025; Li et al., 2024) and to assess their cultural knowledge on specific domains (AlKhamissi et al., 2024; Alwajih et al., 2025a). Beyond cultural adaptation, progress in Arabic NLP has been supported by benchmarks and resources across a variety of tasks, including machine translation (Akallouch and Fardousse, 2025), named entity recognition (Yahia et al., 2024; Jarrar et al., 2024), and question answering (Mozannar et al., 2019).

In this context, the ***Palmx 2025*** shared task was introduced to evaluate the ability of LLMs to capture Arabic cultural knowledge and to promote the development of culturally aware systems for the Arab world (Alwajih et al., 2025b). It includes two subtasks, General Culture and Islamic Culture, each based on datasets of multiple-choice questions in Modern Standard Arabic (MSA), namely *Palmx-GC* and *Palmx-IC*.

In this paper, we present our participating systems for the General Culture subtask (*Phoenix*) and the Islamic Culture subtask (*PhoenixIs*) of Palmx 2025. Starting from Palmx-GC as seed data, we applied three data augmentation strategies: question paraphrasing, which generates semantically equivalent variants of existing questions, sample-based augmentation, which produces new multiple-choice questions by conditioning on individual question–answer pairs, and dataset-based augmentation, which creates thematically related questions by leveraging the full dataset (Section 4.1). For the Islamic Culture subtask, we only explored question paraphrasing. The augmented data was then used to fine-tune dedicated models for each subtask. Our experiments demonstrate that the proposed augmentation strategies improve performance on both subtasks. Phoenix obtains an accuracy of 71.35% on the General Culture subtask, and PhoenixIs attains 83.82% on the Islamic Culture subtask.

Our contributions in this work are: **(1)** we present Phoenix and PhoenixIs, two systems developed for the General Culture and Islamic Culture subtasks, respectively. **(2)** We design and evaluate three data augmentation strategies that enrich the available training data and improve model perfor-

[*] Equal Contribution.

mance. **(3)** We provide an extensive analysis of these strategies, showing that they enhance accuracy on both subtasks.

2 Related Work

Although large language models have achieved strong performance across a variety of languages, adapting them to specific cultural contexts, particularly those that are low-resource, remains a significant challenge, as they often display a bias toward Western culture (Cecilia Liu et al., 2024; Navigli et al., 2023; Cao et al., 2023; Naous et al., 2024). To mitigate this issue, several adaptation strategies have been explored, including continuous pretraining (Mekki et al., 2025), prompt tuning (Masoud et al., 2024), prompt engineering (Shen et al., 2024; Tao et al., 2024; AlKhamissi et al., 2024), and supervised fine-tuning (Li et al., 2024). All of these approaches rely, to varying degrees, on the availability of well-constructed cultural datasets, which remain scarce. In response, a growing body of work has focused on building resources that capture cultural knowledge across different languages and communities (Alwajih et al., 2025a; Myung et al., 2024).

Since the manual annotation of cultural data is resource-intensive and difficult to scale, researchers have increasingly turned to data augmentation to expand training sets (Liu et al., 2025; Li et al., 2024; Joshi et al., 2024). Nonetheless, this approach requires careful design to ensure that the generated data maintains quality and reliability (Liu et al., 2024).

The limited representation of Arabic in pretraining corpora has motivated a growing effort to develop LLMs specifically designed for the Arab world. One approach has been to rely on translation, where large volumes of English data are translated into Arabic to supplement training resources (Sengupta et al., 2023). Other work has emphasized the inclusion of native Arabic data without translation in order to better capture the linguistic and cultural features of the language (Huang et al., 2024). To address the bias toward English and the resulting cultural misalignment, some approaches have relied on continual pretraining with carefully curated cultural data (Mekki et al., 2025), while others have explored training models entirely from scratch (Bari et al., 2024; Team et al., 2025). In this work, we build on these efforts by finetuning such models for the Palmx shared task subtasks.

3 Palmx

The Palmx shared task was established to evaluate the ability of LLMs to capture Arabic cultural knowledge and to encourage the creation of systems that are culturally aware within the Arab world. It is divided into two subtasks: General Culture and Islamic Culture. The General Culture subtask examines the ability of LLMs to reason about different aspects of Arabic culture. Its questions span a wide range of domains such as customs, etiquette, and arts from across Arab countries, including Palestine, Morocco, Egypt, and others. The Islamic Culture subtask is designed to evaluate models' understanding of central elements of Islamic culture. The questions address topics including religious practices, Quranic knowledge, and Hadith literature.

3.1 Datasets

The Palmx shared task provides two datasets, Palmx-GC for the General Culture subtask and Palmx-IC for the Islamic Culture subtask. Both datasets consist of multiple-choice questions in MSA and are split into training, development, and blind test sets. Table 1 presents the detailed statistics for each split.

Split	Palmx-GC	Palmx-IC
Training	2000	600
Development	500	300
Blind Test	1000	1000

Table 1: Distribution of samples in Palmx-GC and Palmx-IC.

4 Phoenix

We propose two systems for the Palmx shared task: *Phoenix* for the General Culture subtask and *PhoenixIs* for the Islamic Culture subtask. Both systems build on the official provided datasets, and each incorporates data augmentation to expand the training data before finetuning task-specific models. An overview of the augmentation strategies is presented in Figure 1.

4.1 Data Augmentation

4.1.1 Question Paraphrasing

In the paraphrasing setup, the LLM was provided with a single question and instructed to generate n_1 semantically equivalent variants. This strategy

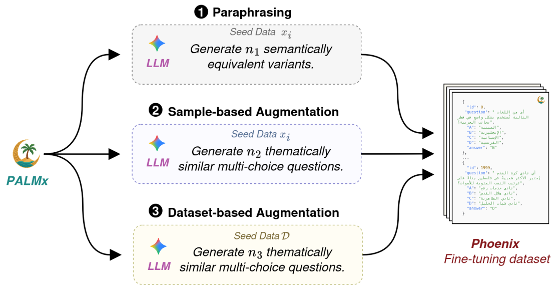

❶ Paraphrasing
Seed Data x_i
LLM
Generate n_1 semantically equivalent variants.
❷ Sample-based Augmentation
Seed Data x_i
LLM
Generate n_2 thematically similar multi-choice questions.
❸ Dataset-based Augmentation
Seed Data $\mathcal{D}$
LLM
Generate n_3 thematically similar multi-choice questions.
PALMx
Phoenix
Fine-tuning dataset

Figure 1: Overview of Phoenix data augmentation strategies

increases data diversity while preserving the original meaning, thereby helping the model generalize to different phrasings of the same cultural concept. We employed Gemini 2.5 Pro for this augmentation.

4.1.2 Sample-based Augmentation

In sample-based augmentation, the LLM was given an original question together with its multiple-choice answers and asked to generate new thematically and structurally similar multiple-choice questions. This approach expands the dataset by producing additional questions that maintain the original format while introducing controlled variation. We used Gemini 2.5 Flash for sample-based augmentation.

4.1.3 Dataset-based Augmentation

For dataset-based augmentation, the LLM was provided with the full Palmx-GC dataset of 2,000 samples and prompted to generate n_3 new multiple-choice questions that are thematically related. Unlike the previous strategies, this method leverages the dataset as a whole, encouraging the model to create questions that capture broader patterns across domains. Gemini 2.5 Pro was used for this setup. To ensure quality and cultural fidelity, a random subset of the LLM-generated questions from each augmentation strategy was manually inspected by human annotators. This verification step helped confirm semantic correctness and adherence to cultural context before including the data in training (see Appendix B for a detailed error analysis).

4.2 Finetuning Data

For finetuning, we constructed task-specific datasets by combining the original seed data with the augmented questions. In the General Culture subtask, *Phoenix* was finetuned on a total of 18,742 questions, consisting of 2,000 from Palmx-GC, 6,000 from question paraphrasing, 6,411 from sample-based augmentation, and 4,331 from dataset-based augmentation. For the Islamic Culture subtask, *PhoenixIs* was finetuned on 4,400 questions, including 600 from Palmx-IC, 1,800 from question paraphrasing, and 2,000 from Palmx-GC.

4.3 Model Pre-selection

To identify suitable base models for finetuning, we first evaluated several state-of-the-art Arabic-focused LLMs in a zero-shot setting on the Palmx-GC and Palmx-IC validation sets. The results are summarized in Table 2. Based on this evaluation, we selected *Fanar-1-9B-Instruct* (Team et al., 2025) for *Phoenix* and *ALLaM-7B-Instruct* (Bari et al., 2024) for *PhoenixIs*.

Model	Accuracy (%)
Category: General Culture (GC)	
Fanar-1-9B-Instruct	**72.40**
ALLaM-7B-Instruct	70.60
NileChat-3B	70.00
AceGPT-v2-8B-Chat	65.00
Falcon-H1-7B-Instruct	39.20
Category: Islamic Culture (IC)	
ALLaM-7B-Instruct	**73.00**
Fanar-1-9B-Instruct	67.00
NileChat-3B	64.00
AceGPT-v2-8B-Chat	63.67
Falcon-H1-7B-Instruct	34.00

Table 2: Zero-shot accuracy of different LLMs on the Palmx-GC (GC) and Palmx-IC (IC) validation sets. The top-performing model in each category is highlighted.

5 Experiment and Results

5.1 Experimental Setup

Experiments were conducted on the Palmx shared-task datasets for General Culture (GC) and Islamic Culture (IC). Fanar-1-9B-Instruct was selected as the base model for GC, and ALLaM-7B-Instruct for IC, following the pre-selection analysis in Subsection 4.3. Model performance was evaluated using accuracy on the validation and blind test splits. For GC, the fine-tuning corpus combined the original Palmx data with progressively applied augmentation strategies: paraphrasing (PA), sample-based

(SA), and dataset-based (DA). For IC, augmentation was deliberately restricted to paraphrasing in order to safeguard the theological fidelity of religious material. All results are reported on the validation set, with the exception of the official leaderboard scores, which are based on the blind test set. All experiments were repeated three times with different random seeds, and we report the average accuracy. We use Lora (Hu et al., 2022) to finetune both models with a learning rate of 0.0002, an effective batch size of 128, and LoRA hyperparameters $R = 64$, $\alpha = 16$, and dropout 0.1. All experiments were conducted on one NVIDIA A100 GPU.

5.2 Official Leaderboard Results

Table 3 presents the official Palmx 2025 leaderboard. Phoenix achieved fourth place in GC with an accuracy of 71.35%, performing within one percentage point of the leading system. PhoenixIs achieved 83.82% in the Islamic Culture subtask, ranking second among all submitted systems. These results indicate that our augmentation strategies enabled competitive performance across both subtasks.

Rank	Team	Accuracy (%)
Category: General Culture (GC)		
1	HAI	72.15
2	Pulkit Chatwal	71.65
3	AYA_Team	71.45
4	**Phoenix (ours)**	**71.35**
5	CultranAI	70.50
6	ISL-NLP	67.60
7	Rafiul Biswas	67.55
8	Hamyaria	65.90
9	Star	64.05
Category: Islamic Culture (IC)		
1	AYA Team	84.22
2	**PhoenixIs (ours)**	**83.82**
3	HAI	82.52
4	Rafiul Biswas	74.13
5	Hamyaria	70.83
6	TarnishedLab	62.84

Table 3: Official Palmx 2025 results. Our team's entries are highlighted.

Fine-tuning data	Acc.
General Culture (GC)	
Palmx	77.73 ± 1.21
Palmx + PA	80.07 ± 1.21
Palmx + PA + SA	80.60 ± 1.06
Palmx + PA + SA + DA	$\mathbf{80.93 \pm 0.76}$
Islamic Culture (IC)	
Palmx Islamic	73.73 ± 2.92
Palmx Islamic + PA	75.11 ± 1.91
Palmx Islamic + PA + Palmx-GC	$\mathbf{78.56 \pm 0.78}$

Table 4: Ablation on validation sets (mean ± std over three runs). GC uses Fanar-1-9B-Instruct; IC uses ALLaM-7B-Instruct.

5.3 Ablation Study: Impact of Augmentation

To assess the contribution of each augmentation strategy, controlled ablations were performed on the validation sets (Table 4). In GC, performance improved consistently with each additional augmentation step, reaching 80.93% with the full combination of PA, SA, and DA. This trend illustrates that diversity introduced at both the question and dataset level substantially enhances generalization. In IC, we explored the effect of paraphrasing and the inclusion of Palmx-GC in the finetuning mixture. The base model achieved 73.73% accuracy. Incorporating paraphrasing increased performance to 75.11%, and adding Palmx-GC further raised accuracy to 78.56%. Overall, our study shows that each component of the proposed augmentation strategy contributed to the final performance.

6 Conclusion

In this work, we presented Phoenix and PhoenixIs, two systems developed for the Palmx 2025 shared task on Arabic cultural understanding. By leveraging the Palmx-GC and Palmx-IC datasets and applying a range of data augmentation strategies, we constructed enriched fine-tuning sets. Our experiments showed that our proposed data augmentation strategies enabled consistent improvements across both General Culture and Islamic Culture subtasks.

Limitations.

Our approach relies on synthetic augmentation for the General Culture task, which, while effective, may introduce distributional biases or artifacts. Human verification was applied to sampled augmented data, but the majority of generated content remained unreviewed. For the Islamic Culture task, augmentation was deliberately restricted to paraphrasing to preserve theological fidelity, which limited the exploration of richer augmentation strategies.

References

Oussama Akallouch and Khalid Fardousse. 2025. In-context learning for low-resource machine translation: A study on tarifit with large language models. *Algorithms*, 18(8):489.

Badr AlKhamissi, Muhammad ElNokrashy, Mai Alkhamissi, and Mona Diab. 2024. Investigating cultural alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12404–12422, Bangkok, Thailand. Association for Computational Linguistics.

Fakhraddin Alwajih, Abdellah El Mekki, Samar Mohamed Magdy, AbdelRahim A. Elmadany, Omer Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, Abdulfattah Mohammed Yahya, Rahaf Alhamouri, Hamzah A. Alsayadi, Hiba Zayed, Sara Shatnawi, Serry Sibaee, Yasir Ech-chammakhy, Walid Al-Dhabyani, Marwa Mohamed Ali, Imen Jarraya, and 25 others. 2025a. Palm: A culturally inclusive and linguistically diverse dataset for Arabic LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32871–32894, Vienna, Austria. Association for Computational Linguistics.

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025b. PalmX 2025: The First Shared Task on Benchmarking LLMs on Arabic and Islamic Culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan Al-Rashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, and 1 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing cross-cultural alignment between chatgpt and human societies: An empirical study. *arXiv preprint arXiv:2303.17466*.

Chen Cecilia Liu, Fajri Koto, Timothy Baldwin, and Iryna Gurevych. 2024. Are multilingual LLMs culturally-diverse reasoners? an investigation into multicultural proverbs and sayings. 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 2016–2039, Mexico City, Mexico. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Song Dingjie, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 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 (Volume 1: Long Papers)*, pages 8139–8163, Mexico City, Mexico. Association for Computational Linguistics.

Mustafa Jarrar, Nagham Hamad, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2024. Wojoodner 2024: The second arabic named entity recognition shared task. *arXiv preprint arXiv:2407.09936*.

Raviraj Joshi, Kanishk Singla, Anusha Kamath, Raunak Kalani, Rakesh Paul, Utkarsh Vaidya, Sanjay Singh Chauhan, Niranjan Wartikar, and Eileen Long. 2024. Adapting multilingual llms to low-resource languages using continued pre-training and synthetic corpus. *arXiv preprint arXiv:2410.14815*.

Raviraj Joshi, Kanishk Singla, Anusha Kamath, Raunak Kalani, Rakesh Paul, Utkarsh Vaidya, Sanjay Singh Chauhan, Niranjan Wartikar, and Eileen Long. 2025. Adapting multilingual LLMs to low-resource languages using continued pre-training and synthetic corpus: A case study for Hindi LLMs. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 50–57, Abu Dhabi. Association for Computational Linguistics.

Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024. Culturellm: Incorporating cultural differences into large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 84799–84838. Curran Associates, Inc.

Chen Cecilia Liu, Anna Korhonen, and Iryna Gurevych. 2025. Cultural learning-based culture adaptation of language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3114–3134, Vienna, Austria. Association for Computational Linguistics.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi

Yang, Denny Zhou, and 1 others. 2024. Best practices and lessons learned on synthetic data. *arXiv preprint arXiv:2404.07503*.

Reem I Masoud, Martin Ferianc, Philip Colin Treleaven, and Miguel RD Rodrigues. 2024. Llm alignment using soft prompt tuning: The case of cultural alignment. In *Workshop on Socially Responsible Language Modelling Research*.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities. *arXiv preprint arXiv:2505.18383*.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural Arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118, Florence, Italy. Association for Computational Linguistics.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, Víctor Gutiérrez-Basulto, Yazmín Ibáñez García, Hwaran Lee, Shamsuddeen Hassan Muhammad, Kiwoong Park, Anar Sabuhi Rzayev, Nina White, Seid Muhie Yimam, Mohammad Taher Pilehvar, and 3 others. 2024. Blend: A benchmark for llms on everyday knowledge in diverse cultures and languages. In *Advances in Neural Information Processing Systems*, volume 37, pages 78104–78146. Curran Associates, Inc.

Tarek Naous, Michael J Ryan, Alan Ritter, and Wei Xu. 2024. Having beer after prayer? measuring cultural bias in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16366–16393, Bangkok, Thailand. Association for Computational Linguistics.

Roberto Navigli, Simone Conia, and Björn Ross. 2023. Biases in large language models: origins, inventory, and discussion. *ACM Journal of Data and Information Quality*, 15(2):1–21.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Siqi Shen, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, Soujanya Poria, and Rada Mihalcea. 2024. Understanding the capabilities and limitations of large language models for cultural commonsense. 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 5668–5680, Mexico City, Mexico. Association for Computational Linguistics.

Yan Tao, Olga Viberg, Ryan S Baker, and René F Kizilcec. 2024. Cultural bias and cultural alignment of large language models. *PNAS nexus*, 3(9):pgae346.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, and 1 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Issam Yahia, Houdaifa Atou, and Ismail Berrada. 2024. Addax at WojoodNER 2024: Attention-based dual-channel neural network for Arabic named entity recognition. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 867–873, Bangkok, Thailand. Association for Computational Linguistics.

A Effectiveness Analysis

We investigated the effect of increasing the size of augmented data on model performance, as generating too many samples can hurt the performance. For Phoenix, we fine-tuned the model with 2,000, 8,000, 15,000, and 18,742 samples, with results shown in Figure 2. The best performance was achieved with 18,742 samples. Similarly, for PhoenixIs, we fine-tuned with 2,600, 3,200, 3,800, and 4,400 samples, as shown in Figure 3, where 4,400 samples yielded the strongest results. The composition of each set is detailed in Tables 5 and 6.

	S1	**S2**	**S3**	**S4**
Palmx-GC	2,000	2,000	2,000	2,000
PA	0	2,000	4,000	6,000
SA	0	2,000	4,000	6,411
DA	0	2,000	3,000	4,331
Total	2,000	8,000	13,000	18,742

Table 5: Composition of the dataset for each experiment on Phoenix.

	S1	**S2**	**S3**	**S4**
Palmx-IC	600	600	600	600
Palmx-GC	2000	2000	2000	2000
PA	0	600	1200	1800
Total	2600	3200	3800	4400

Table 6: Composition of the dataset for each experiment on PhoenixIs.

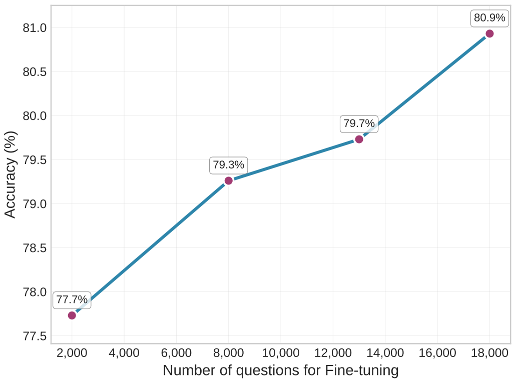


Figure 2: Influence of the number of fine-tuning samples on Phoenix.

B Human Verification and Error Analysis

To ensure the reliability of the augmented data, we conducted a manual verification of randomly sampled questions from each augmentation strategy:

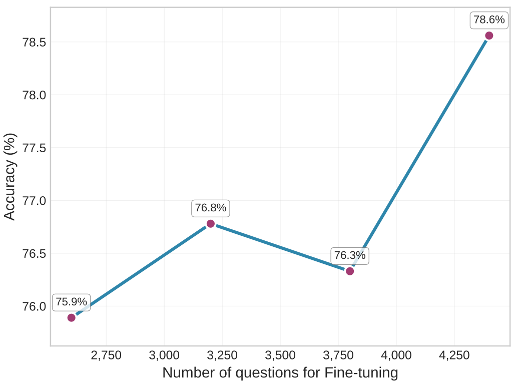
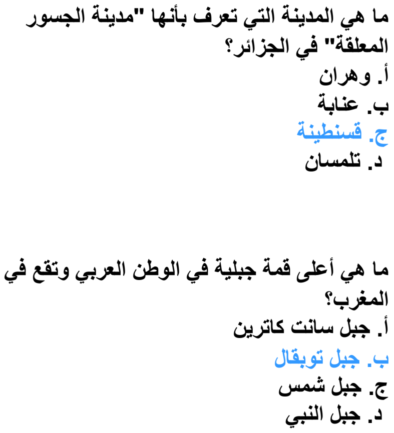


Figure 3: Influence of the number of fine-tuning samples on PhoenixIs.

- **Sample-based Augmentation (SA):** From a random set of 100 generated questions, we identified **10 problematic cases**. Of these, **3 were factually incorrect**, while the remaining **7 deviated from instructions** (e.g., not strictly following the required format or asking about tangential topics). Importantly, most of these still produced valid question–answer pairs despite the inconsistencies.
- **Dataset-based Augmentation (DA):** From a random set of 100 generated questions, we found **3 issues**, all of which were culturally valid but referenced **non-Arab countries**.

Overall, the error rate across both strategies was relatively low. The main sources of error were format deviation and domain drift rather than factual inaccuracies. This indicates that our augmentation pipeline is broadly reliable.


ما هي المدينة التي تعرف بأنها "مدينة الجسور
المعلقة" في الجزائر؟
أ. وهران
ب. عنابة
ج. قسنطينة
د. تلمسان

ما هي أعلى قمة جبلية في الوطن العربي وتقع في
المغرب؟
أ. جبل سانت كاترين
ب. جبل توبقال
ج. جبل شمس
د. جبل النبي

Figure 4: Cases from our generated data where the generation was correct. The proposed answer is highlighted in blue.

يُعرف الفنان السوداني الكبير محمد وردي بلقب 'فنان
أفريقيا الأول'. ما هو اللقب الآخر الذي ارتبط به بشكل وثيق
ويعكس مكانته في الغناء السوداني؟
أ. بلبل السودان
ب. عنقريب الفن
ج. فنان الشعب
د. فنان الوادي

ما هو أعلى جبل في اليابان؟
أ. جبل كيتا
ب. جبل هوتاكا
ج. جبل فوجي
د. جبل أينو

Figure 5: Cases from our generated data where the generation was incorrect (or deviated from instructions). The proposed answer is highlighted in blue.

C Error Analysis on Validation Set

We inspected a small sample of incorrect validation set predictions to illustrate typical failure cases. Figure 6 shows four representative errors, where the ground truth is marked in green and the model's predictions are in red.

ما هي الدولة التي تحتل المرتبة الثالثة في مساحة
الأراضي في إفريقيا والعالم العربي؟
أ. السودان
ب. الجزائر
ج. نيجيريا
د. مصر

متى يُحتفل بيوم الرياضة الوطني في قطر؟
أ. في الثلاثاء من الأسبوع الثاني من فبراير
ب. في 1 مايو
ج. في 10 ذو الحجة
د. في 18 ديسمبر

أي من الشخصيات التالية يُعتبر روائياً ليبياً؟
أ. هشام مطر
ب. محمد الفيتوري
ج. جمال خشتة
د. عزالدين شكري الفيلالي

ما هو الدين الذي يعتنقه النوبيون والبجا بشكل تقليدي في
السودان؟
أ. المسيحية
ب. اليهودية
ج. الهندوسية
د. الإسلام

Figure 6: Cases from the Palmx-GC validation set where the model's prediction was incorrect. Ground truth is marked in green, predictions in red.

Paraphrasing

من أي لونين تتألف الكوفية الفلسطينية؟
أ. أزرق وذهبي
ب. أحمر وأخضر
ج. أبيض وأسود
د. بني ورمادي

Sample-based Augmentation

من هو الكاتب الفلسطيني المعروف بقصصه
القصيرة التي تناولت القضية الفلسطينية وعُرفت
بأسلوبها الواقعي والرمزي؟
أ. محمود درويش
ب. سميح القاسم
ج. إبراهيم نصر الله
د. غسان كنفاني

Dataset-based Augmentation

أي من هذه الأنهار يشكل الحدود بين الأردن
وفلسطين؟
أ. نهر اليرموك
ب. نهر الليطاني
ج. نهر الأردن
د. نهر العاصي

Figure 7: Examples of augmented question–answer pairs generated using the three strategies: paraphrasing, sample-based augmentation, and dataset-based augmentation. The proposed answer is highlighted in blue.

QIAS 2025: Overview of the Shared Task on Islamic Inheritance Reasoning and Knowledge Assessment

Abdessalam BOUCHEKIF[1], **Samer RASHWANI**[1], **Emad MOHAMED**[2],
Mutaz AL-KHATIB[1], **Heba SBAHI**[1], **Shahd GABEN**[1], **Wajdi ZAGHOUANI**[3],
Aiman ERBAD[4], **Mohammed GHALY**[1]

[1]**Hamad Bin Khalifa University, Qatar** [2]**Nazarbayev University, Kazakhstan**
[3]**Northwestern University, Qatar** [4]**Qatar University, Qatar**
abouchekif, srashwani, mghaly, malkhatib, sgaben, hsbahi@hbku.edu.qa
emad.mohamed@nu.edu.kz wajdi.zaghouani@northwestern.edu aerbad@qu.edu.qa

Abstract

This paper provides a comprehensive overview of the QIAS 2025 shared task, organized as part of the ArabicNLP 2025 conference and co-located with EMNLP 2025. The task was designed for the evaluation of large language models in the complex domains of religious and legal reasoning. It comprises two subtasks: *(1)* Islamic Inheritance Reasoning, requiring models to compute inheritance shares according to Islamic jurisprudence, and *(2)* Islamic Knowledge Assessment, which covers a range of traditional Islamic disciplines. Both subtasks were structured as multiple-choice question answering challenges, with questions stratified by varying difficulty levels. The shared task attracted significant interest, with 44 teams participating in the development phase, from which 18 teams advanced to the final test phase. Of these, 6 teams submitted entries for both subtasks, 8 for Task 1 only, and two for Task 2 only. Ultimately, 16 teams submitted system description papers. Herein, we detail the task's motivation, dataset construction, evaluation protocol, and present a summary of the participating systems and their results.

1 Introduction

The emergence of Large Language Models (LLMs) has transformed NLP, enabling state-of-the-art performance in tasks requiring deep linguistic understanding, complex reasoning, and coherent text generation. Trained on large-scale general-purpose corpora, LLMs have demonstrated strong performance across a variety of benchmarks, including question answering, summarization, and dialogue. However, LLMs still face challenges in specialized domains, particularly those requiring high information accuracy, and sensitivity to cultural or religious contexts. In the Islamic contexts, LLMs must reason over authoritative and structured sources such as the Qur'an, Hadith, and fatwas. They must also consider differences in interpretation across schools of thought, including variations within Sunni Islam across the four major legal schools: Ḥanafī, Mālikī, Shāfiʿī, and Ḥanbalī.

To evaluate LLMs' capabilities in both Islamic legal reasoning and specialized religious knowledge, we introduce the QIAS 2025 Shared Task. This benchmark presents a diverse set of question-answering challenges across multiple domains, difficulty levels, and jurisprudential perspectives. The task includes two subtasks: (1) Islamic Inheritance Reasoning, which requires precise, rule-based reasoning grounded in classical Islamic jurisprudence. Task 2 focuses on general Islamic knowledge, incorporating questions curated by experts from key disciplines. Each question is labeled by difficulty and assesses knowledge of religious concepts, legal reasoning, and interpretive differences.

In this paper, we present an overview of the QIAS 2025[1] Shared Task, which represents an important step toward developing NLP models capable of addressing complex challenges in Islamic knowledge. This includes inheritance calculation tasks requiring precise reasoning and rule-based computation grounded in Islamic jurisprudence. To our knowledge, no previous dataset has been specifically designed for fine-tuning models on Islamic inheritance reasoning at this scale. The second task focuses on question answering covering diverse areas of Islamic scholarship. Unlike many existing datasets relying on general cultural or surface-level questions, our dataset is curated and annotated by domain experts to reflect a deeper understanding of jurisprudential and theological concepts.

2 Related Work

Recent LLMs such as GPT-4 (Achiam et al., 2023), Gemini2.5 (Comanici et al., 2025), and DeepSeek-R1 (Guo et al., 2025) have achieved state-of-the-art performance across diverse stan-

[1] https://sites.google.com/view/qias2025/

Proceedings of The Third Arabic Natural Language Processing Conference, pages 851–860
November 8-9, 2025 ©2025 Association for Computational Linguistics

dard NLP benchmarks. In parallel, several Arabic-focused LLMs have been developed to better capture linguistic, cultural, and domain-relevant needs of Arabic-speaking communities, including Falcon(Almazrouei et al., 2023), Jais (Sengupta et al., 2023), AceGPT(Huang et al., 2023), ArabianGPT (Koubaa et al., 2024), ALLaM (Bari et al., 2024), and Fanar (Abbas et al., 2025). These efforts have motivated growing interest in applying LLMs to tasks involving Islamic content and knowledge.

The application of LLMs to Islamic texts has recently gained increasing attention within the NLP community. (Malhas et al., 2022) (Malhas et al., 2023) organized shared tasks focused on advancing Islamic information retrieval, with a particular focus on understanding Qur'anic passages. These tasks included a Qur'anic passage retrieval task—requiring models to retrieve relevant verses from the Qur'an given a question, and a reading comprehension task, where expected to extract accurate answers from a provided passage. More recently, (Sayeed et al., 2025) explored QA systems for *ibb nabawī* (Prophetic medicine) using LLaMA-3, Mistral-7B, and Qwen-2 combined with RAG, while (Alan et al., 2024) proposed MufassirQAS, a RAG-based system trained on Turkish Islamic texts to improve transparency and reduce hallucinations in religious QA. (Rizqullah et al., 2023) introduced QASiNa QA dataset, derived from *Sirah Nabawiyah* texts in Indonesian, comparing traditional multilingual transformers (XLM-R, mBERT, IndoBERT) with GPT-3.5 and GPT-4. (Qamar et al., 2024) introduced a dataset of 73,000 question–answer pairs has been introduced, focusing on non-factoid QA for Quranic Tafsir and Hadith. The study revealed a critical gap between automatic evaluation metrics (such as ROUGE) and human judgments. These results show that automatic evaluation metrics alone are not sufficient, and highlight the need for more robust evaluation methods that can better reflect the complexity and interpretive nature of Islamic religious texts. In (Aleid and Azmi, 2025), the authors released Hajj-FQA, a benchmark of 2,826 QA pairs extracted from 800 expert-annotated fatwas concerning the Hajj pilgrimage. Despite these efforts, several studies have identified significant limitations in this LLMs. For instance, (Mohammed et al., 2025) show that even advanced models like GPT-4 tend to produce factually incorrect or misleading answers when applied to Islamic content. They identify three main issues: *(i)* misinterpretation of religious context, *(ii)* generation of answers that are unclear or not based on reliable Islamic sources like the Qur'an or Hadith, and *(iii)* high sensitivity to slight variations in question phrasing, leading to inconsistent responses. Similarly, (Alnefaie et al., 2023) observed that GPT-4 has difficulty answering Quranic questions accurately, due to difficulties with classical arabic, semantic ambiguity, and misinterpretation of contextual meaning.

Early research on automating Islamic inheritance began with expert systems focused on calculating basic inheritance shares (Akkila and Naser, 2016). Later works incorporated intricate adjustments such as *ḥajb*, ʿawl, and radd (Tabassum et al., 2019). (Zouaoui and Rezeg, 2021) proposed a Arabic ontology for identifying heirs and d calculating their inheritance shares (Tabassum et al., 2019). Most recently, (Bouchekif et al., 2025) evaluated seven LLMs on Islamic inheritance. The results reveal that models with strong reasoning capabilities, such as Gemini 2.5 and o3, achieved high performance, with accuracy rates of 90.6% and 93.4%, respectively. In contrast, models lacking advanced reasoning abilities—such as Jais, Mistral, and LLaMA—performed significantly worse, with accuracy rates below 50%, highlighting their limitations in handling complex legal reasoning tasks.

3 Task1: Islamic Inheritance Reasoning

3.1 Task Description

The task1 focuses on the domain of *'lm* al-mawārīth, the Islamic science of inheritance. The goal is to assess the ability of LLMs to accurately apply Islamic inheritance rules in realistic scenarios. Solving inheritance problems requires a combination of cognitive, legal, and computational skills, including:

1. Identifying familial relationships and considering legal conditions such as debts, bequests, and the sequence of deaths among relatives.
2. Determining eligible heirs, including fixed-share heirs (*aṣḥāb al-furūḍ*) and residuaries (*'aṣabāt*), and correctly applying exclusion rules (*ḥajb*) based on valid justifications and authentic scriptural evidence.
3. Computing shares by deriving a common denominator and adjusting the distribution when necessary:
   - *Radd* (redistribution) is used when a surplus remains after initial allocation. This surplus is proportionally redistributed among

the heirs, excluding spouses. — *Example:* Wife $(1/4)$ and full sister $(1/2)$, leaving a surplus of $1/4$; after redistribution, the wife receives $(1/4)$ and the sister receives $(3/4)$.
   - ʿAwl (proportional reduction) is applied when the sum of assigned shares exceeds the estate. All shares are scaled down proportionally. — *Example:* Father $(1/6)$, mother $(1/6)$, wife $(1/8)$, and four daughters $(2/3)$; the total exceeds 1. The denominator is adjusted to 27, and then the wife receives $3/27 = 1/9$.

4. Addressing complex and exceptional cases, such as consecutive death (*munāsakha*) or juristic disputes like the *Akdariyya* case involving grandparents and siblings.
5. Numerical precision in the final distribution, including the correct adjustment and fractional allocation[2].

3.2 Data

The dataset contains 22,000 MCQs, including 10,446 generated from IslamWeb fatwas and 11,554 constructed from inheritance case resolutions using the calculator of the *Almwareeth* website[3], offers a specialized tool that algorithmically solves all types of mirath (Islamic inheritance) problems. The IslamWeb-based MCQs were derived from Islamic religio-ethical rulings (fatwas)[4], which were automatically converted into question-answer format using Gemini 2.5 Pro. Each generated question was then reviewed by four experts in Islamic studies to ensure both legal soundness and linguistic clarity. As part of the preprocessing phase, ambiguous questions were rephrased to guarantee a single, unambiguous interpretation. The answer choices were also revised to eliminate semantic and numerical redundancies, such as equivalent options (*e.g* $1/2$ and $2/4$). The dataset has two levels of difficulty: **Beginner** and **Advanced**, reflecting increasing complexity in both legal reasoning and mathematical computation.

Participants are also provided a collection of 3,165 fatwas (question–answer pairs) from IslamWeb is available. These fatwas cover a broad spectrum of Islamic legal, ethical, and social issues and can serve as a valuable supplementary knowledge base.

Example – Level Beginner

توفي عن أب، و2 أخ شقيق، و1 ابن أخ شقيق، و2 عم شقيق للأب، وأم، و2 بنت، و1 زوجة، ما هو نصيب الأم؟
He was survived by his father, two full brothers, one nephew (son of a full brother), two paternal uncles, his mother, two daughters, and his wife. What is the share of the mother?

(One-third) الثلث □
(One-quarter) الربع □
(One-sixth) السدس ■
(One-eighth) الثمن □
(One-half) النصف □
(Nothing) لا شيء □

Example – Level Advanced

توفي عن زوجة وبنتين وأخ شقيق، والتركة 12000 درهم. ما هو النصيب النهائي لكل وارث من التركة؟
He was survived by his wife, two daughters, and one full brother. The estate is 12,000 dirhams. What is the final share of each heir from the estate?

■ الزوجة: 1500 درهم، البنتان: 8000 درهم، الأخ الشقيق: 2500 درهم
Wife: 1500 dirhams, Two daughters: 8000 dirhams, Full brother: 2500 dirhams

□ الزوجة: 3000 درهم، البنتان: 8000 هم، الأخ الشقيق: 1000 درهم
Wife: 3000 dirhams, Two daughters: 8000 dirhams, Full brother: 1000 dirhams

□ الزوجة: 1500 درهم، البنتان: 6000 درهم، الأخ الشقيق: 4500 درهم
Wife: 1500 dirhams, Two daughters: 6000 dirhams, Full brother: 4500 dirhams

□ الزوجة: 1500 درهم، البنتان: 8000 درهم، الأخ الشقيق: 3000 درهم
Wife: 1500 dirhams, Two daughters: 8000 dirhams, Full brother: 3000 dirhams

□ الزوجة: 2000 درهم، البنتان: 7500 درهم، الأخ الشقيق: 2500 درهم
Wife: 2000 dirhams, Two daughters: 7500 dirhams, Full brother: 2500 dirhams

□ الزوجة: 1000 درهم، البنتان: 8500 درهم، الأخ الشقيق: 2500 درهم
Wife: 1000 dirhams, Two daughters: 8500 dirhams, Full brother: 2500 dirhams

4 Task2: Islamic Assessment

4.1 Task Description

The task2 evaluates general Islamic knowledge across a wide range of topics within Islamic knowledge, including ʿulūm al-Qurʾān (Quranic studies), ʿulūm al-Ḥadīth (hadith criticism), *fiqh* (jurisprudence), uṣūl al-fiqh (legal theory), *sīrah* (Prophetic Biography). It is organized into three progressively

[2] For more details about the terminology and rules of Islamic inheritance law, see "*Irth*," in *Al-Mawsūʿa al-Fiqhiyya* (The Kuwaitan Encyclopedia of Fiqh). Kuwait: *Wazārat al-Awqāf* wa-al-Shuʾūn al-Islamiyya. 45 Vols. 1984-2007. Vol. 3, Pp. 17-79.

[3] https://almwareeth.com/

[4] https://www.islamweb.net/

Task	Split	Levels			Total
		Beg.	Int.	Adv.	
Task 1					
	Training	10000	—	10000	20000
	Dev	500	—	500	1000
	Test	500	—	500	1000
	Total	**11000**	—	**11000**	**22000**
Task 2					
	Training	—	—	—	—
	Dev	350	175	175	700
	Test	700	150	150	1000
	Total	**1050**	**325**	**325**	**1700**

Table 1: Unified distribution of MCQs across dataset splits and difficulty levels for Task 1 (Inheritance Reasoning) and Task 2 (Islamic Knowledge Assessment). "—" indicates not available.

challenging difficulty levels: beginner, intermediate, and advanced.

4.2 Data

The dataset was constructed from collection of 25 relevant classical Islamic books that are widely recognized by scholars as authoritative. It consists of 1,400 MCQs (700 for training and 700 for testing), all rigorously reviewed and validated by five experts in Islamic studies. Each question has been carefully designed to elicit a single, unambiguous correct answer, thereby ensuring clarity and consistency in the evaluation process.
The answers to the MCQs in the validation and test sets are derived from a selection of classical Islamic texts, which we provide to participants. As such, this corpus can be leveraged either as part of a Retrieval-Augmented Generation (RAG) system to enhance the model's ability to generate accurate and contextually grounded responses, or to fine-tune language models on Islamic studies.

Example of MCQ Level Beginner

ما مدة المسح على الخفين للمقيم؟
What is the duration of wiping over the leather socks for a resident?

One day and one night	يوم وليلة ■
Three days and their nights	ثلاثة أيام بلياليهن □
Two days and two nights	يومان وليلتان □
A full week	أسبوع كامل □

Example of MCQ Level Intermediate

من شروط الأصل في القياس؟

Which of the following is a condition for the base case (al-aṣl) in analogical reasoning (*qiyās*)?

□ أن يكون الأصل فرعًا لأصلٍ آخر
That the base case (al-aṣl) is itself a branch (farʾ) of another base case.

□ ألا يكون الحكم ثابتًا في الأصل بطريقٍ سمعيٍّ شرعي
That the ruling in the base case is *not* established by a revealed textual proof.

■ ألا يكون الأصلُ فرعًا لأصلٍ آخر
That the base case (al-aṣl) is *not* a branch (farʾ) of another base case.

□ ألا تُعرَف طريقةُ الاستنباط
That the method of derivation is unknown.

Example of MCQ Level Advanced

ما هو طريق الحكماء لإثبات وجود الواجب؟

What is the method of the philosophers to prove the existence of the necessary Being (*al-Wājib*)?

□ عن طريق اعتبار العالم قديمًا.
By positing the world as eternal.

□ عن طريق إثبات أن العالم واجب لذاته.
By claiming the world is necessary in itself.

■ عن طريق امتناع التسلسل والدور.
By the impossibility of infinite regress (*tasalsul*) and circular causation.

□ عن طريق إثبات حدوث العالم.
By demonstrating that the world is originated.

Team Name	Task 1	Task 2	Affiliations
Gumball (Elrefai et al., 2025)	✔	✔	Alexandria University, Ain Shams University, Benha University
PuxAI (Phuc and Đặng Văn, 2025)	✔	✔	VNU□HCM University of Information Technology
NYUAD (AlDahoul and Zaki, 2025)	✔		New York University Abu Dhabi
HIAST (Hamed et al., 2025)	✔	✔	Higher Institute for Applied Sciences and Technology
MorAI (R'baiti et al., 2025)	✔		Mohammed VI Polytechnic University
CVPD (Bekhouche et al., 2025)	✔		University of the Basque Country, Sorbonne University Abu Dhabi
QU-NLP (AL-Smadi, 2025)	✔	✔	Qatar University
CIS-RG (Zaki et al., 2025)	✔		Sinai University
ANLPers (Sibaee et al., 2025)	✔	✔	Prince Sultan University
Athar (Noureldien et al., 2025)	✔	✔	University of Khartoum, University Malaysia
SHA (Altammami, 2025)	✔		King Saud University
SEA (Alowaidi et al., 2025)	✔		University of Leeds
HAI (Hossain and Afli, 2025)	✔		ADAPT Centre
IWAN	✔		King Saud University
Transform_Tafsir (Abu Ahmad et al., 2025)	✔		University of Osnabrück, German Research Center for Artificial Intelligence
N&N (Alangari and AlShenaifi, 2025)		✔	King Saud University
Teams60		✔	MBZUAI
Tokenizers United (Samy et al., 2025)		✔	Nile University, Ain Shams University

Table 2: The participating teams: tasks and affiliations.

5 Results and Discussion

A total of 17 teams participated in the Test phase. Among these, 6 teams submitted systems for both subtasks, 7 teams participated in Task 1 only, and 2 teams in Task 2 only. Table 2 summarizes the participating teams and their affiliations. The Dev phase lasted approximately one and a half months, followed by a 5-day test phase. During the test phase, participants made a total of 127 submissions for Task 1 and 50 for Task 2. We use accuracy to evaluate models, calculated as the percentage of questions for which the model's prediction exactly matches the correct answer. We provide a baseline implementation using Fanar, 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. [5]

5.1 Participating Teams and Results

Table 3 presents the leaderboard rankings and accuracy scores for both subtasks. In Subtask 1 (Islamic Inheritance Reasoning), the best-performing sys-

[5] https://gitlab.com/islamgpt1/qias_shared_task_2025

Task 1			Task 2		
Rank	Team	Accuracy	Rank	Team	Accuracy
1	Gumball	0.972	1	PuxAI	0.9369
2	PuxAI	0.957	2	Athar	0.9272
3	NYUAD	0.927	3	HIAST	0.9259
4	HIAST	0.895	4	N&N	0.8984
5	MorAI	0.880	5	Tokenizers United	0.8738
6	CVPD	0.876	6	SEA	0.8601
7	QU-NLP	0.859	7	Teams60	0.8491
8	CIS-RG	0.763	8	Transformer_Tafsir	0.7970
9	ANLPers	0.707	9	CIS-RG	0.7874
10	Athar	0.704			
11	SHA	0.624			
12	SEA	0.599			
13	HAI	0.547			
14	Baseline	0.515			
15	IWAN	0.496			
16	Transform_Tafsir	0.447			

Table 3: Accuracy performance of teams on Task 1 and Task 2.

tem reached an accuracy of 97.2%, showcasing strong capabilities in handling complex jurisprudential computations. In Subtask 2 (Islamic Knowledge Assessment), the top score was 93.7%, reflecting the broader challenge of covering multiple Islamic disciplines.

The ***Gumball*** team (Elrefai et al., 2025) secured first place in Subtask 1 with a Qwen3-4B model fine-tuned through a two-stage pipeline combining classical inheritance texts with supervised MCQ training. Their system achieved 97.2% accuracy, outperforming all other submissions.

The ***PuxAI*** team (Phuc and Đặng Văn, 2025), ranked second, introduced a hybrid multi-agent architecture. For inheritance, they developed a *Virtual Inheritance Expert* pipeline combining fatwa retrieval with rule-based reasoning. For general knowledge, they designed a *Proponent–Critic Debate* pipeline, where agents engaged in adversarial reasoning before synthesis. Their system reached 95.7% on Subtask 1 and 93.7% on Subtask 2.

The ***NYUAD*** team (AlDahoul and Zaki, 2025), in third place, evaluated a diverse set of models, including open-source Arabic LLMs (Falcon3, Fanar, Allam), proprietary systems (GPT-4o, GPT-o3, Gemini Flash 2.5, Gemini Pro 2.5), and fine-tuned variants. While Arabic open-source models remained below 40% accuracy, proprietary models achieved up to 92.3%. Their final ensemble system (GPT-o3, Gemini Flash 2.5, Gemini Pro 2.5) reached 92.7%.

The ***HIAST*** team (Hamed et al., 2025) implemented a lightweight RAG pipeline based on Claude 4 Opus, retrieving top-ranked sources (often IslamWeb) and appending them to Arabic few-shot prompts. This approach improved inheritance reasoning, achieving 89.5% accuracy.

The ***MorAI*** team (R'baiti et al., 2025) proposed a collaborative LLM framework combining majority voting with retrieval-augmented generation. Their system integrated ALLaM-7B, DeepSeek-Reasoner, and Gemini-2.5-Flash, each independently generating predictions, with a voting mechanism selecting the final answer. Augmented with TF-IDF retrieval over a curated inheritance case database, their ensemble achieved 88.0% on Subtask 1, compared to 79.5% for ALLaM-7B, 71.8% for DeepSeek-Reasoner, and 83.5% for Gemini-2.5-Flash.

The ***CVPD*** team (Bekhouche et al., 2025) developed an encoder-based approach using Arabic text encoders with an Attentive Relevance Scoring

(ARS) module. Their best configuration, MARBERT with ARS, achieved 69.9% accuracy, while commercial LLMs such as Gemini reached up to 87.6%.

The ***QU-NLP*** team (AL-Smadi, 2025) fine-tuned Fanar-1-9B with LoRA and integrated it into a FAISS-based RAG pipeline. Their system achieved 85.8% accuracy, outperforming GPT-4.5 (74.0%), LLaMA-3 (48.8%), Mistral (44.5%), ALLaM-7B (42.9%), and the Fanar base model (48.1%).

The ***CIS-RG*** team (Zaki et al., 2025) combined fine-tuning, chain-of-thought prompting, and retrieval-augmented generation across multiple models, including Fanar, LLaMA, Gemini, and Mistral. Their hybrid system achieved 76.3% accuracy on Subtask 1, demonstrating competitive reasoning on basic inheritance cases but struggling with complex scenarios such as ʿawl and *ḥajb*.

The ***N&N*** team (Alangari and AlShenaifi, 2025) developed a system based on few-shot chain-of-thought prompting combined with ensemble methods and retrieval-augmented re-prompting (R^{2}P). Their pipeline consisted of (i) few-shot CoT prompting with standardized Arabic templates; (ii) a majority-vote ensemble over GPT-4o, Gemini 2.5, DeepSeek, and Qwen-plus; and (iii) retrieval-augmented re-prompting when the ensemble failed to agree. This design achieved 89.9% accuracy on Subtask 2, ranking them second overall in this task.

The ***ANLPers*** team (Sibaee et al., 2025) focused on Chain-of-Thought prompting, testing Claude 3.7 Sonnet and GPT-4o with direct-answer and step-by-step reasoning. Structured reasoning improved accuracy from 67.0% to 81.0% on Claude 3.7 and from 63.0% to 74.0% on GPT-4o. Error analysis revealed persistent difficulties with *tasheeh* (integer normalization of shares).

The ***Athar*** team (Noureldien et al., 2025) explored both subtasks with distinct strategies. For Subtask 1, they employed a zero-shot DeepSeek-R1 pipeline with constrained prompting and regex-based label extraction, achieving 70.4% accuracy. For Subtask 2, they designed a three-stage hybrid RAG pipeline combining BM25 and dense retrieval with GPT-based reranking, reaching 92.7% and ranking second overall. Their analysis highlighted sensitivity to question length and answer option complexity in inheritance reasoning, and retrieval errors as the main limitation in broader knowledge assessment.

The ***SHA*** team (Altammami, 2025) integrated static and dynamic few-shot prompting with retrieval-augmented generation. Although some models showed performance drops when augmented with additional context, their best configuration—Gemini with RAG and dynamic prompting—achieved 62.3% accuracy.

The ***SEA*** team (Alowaidi et al., 2025) designed an Islamic RAG framework with three stages: (i) knowledge resource preparation, preprocessing fatwas and Islamic books into 500-token chunks indexed in FAISS; (ii) retrieval using similarity search, Keyword-Augmented Two-Stage Retrieval (K2R), or Multi-Query Reformulation (MQR-K); and (iii) answer generation with structured prompting and post-generation validation. Their system achieved 60.0% accuracy on Subtask 1 and 86.0% on Subtask 2.

The ***ADAPT–MTU HAI*** team (Hossain and Afli, 2025) introduced a dual-expert architecture based on ALLaM-7B, combining a LoRA fine-tuned inheritance specialist with its base model. A constrained decoding mechanism enforced valid outputs (A–F). Their system achieved 54.7% accuracy, improving substantially over the 42.9% ALLaM-7B zero-shot baseline.

The ***Transformer Tafsir*** team (Abu Ahmad et al., 2025) developed a hybrid RAG pipeline combining sparse (BM25) and dense retrieval with cross-encoder reranking. While gains in inheritance reasoning were modest (Fanar: 44.0% → 45.0%; Mistral: 35.0% → 39.0%), Subtask 2 showed substantial improvements (Fanar: 55.0% → 80.0%; Mistral: 69.0% → 79.0%).

The ***Tokenizers United*** team (Samy et al., 2025) proposed a Retrieval-Augmented Generation (RAG) pipeline that combined *Muffakir* embeddings for domain-specific retrieval with the Gemini 2.5 Flash Lite model for lightweight generative reasoning. Their design prioritized efficiency, opting for direct similarity search (Top-K = 8–10) rather than complex reranking mechanisms. On the development set, performance varied between 44.3% and 84.3%, depending on the configuration. Their best-performing setup—Qdrant with cosine similarity, a chunk size of 400 characters, and Muffakir embeddings—achieved 87.4% accuracy on the official test set, ranking 5th out of 10 participating teams in Task 2.

6 Conclusions and Future work

In this paper, we presented the QIAS 2025 Shared Task, designed to evaluate the capabilities of large language models in understanding and reasoning within Islamic knowledge domains. The task was divided into two subtasks: Islamic Inheritance Reasoning and Islamic Knowledge Assessment, both formulated as multiple-choice question answering problems with varying levels of difficulty. The submitted systems revealed significant performance gaps between open-source and commercial LLMs, with commercial models showing notably stronger results.

As a future direction, we plan to organize a follow-up edition of the shared task focused more deeply on Islamic inheritance. Unlike the current multiple-choice setup, the next edition will involve end-to-end problem solving—from identifying eligible heirs based on a given scenario to computing their exact shares. This approach will better reflect real-world applications and offer a more rigorous benchmark for legal reasoning tasks. In this context, we will also encourage researchers to use small and open-source language models. These models are easier to deploy, more accessible, and promote better transparency and reproducibility. We hope this will empower researchers—especially in low-resource settings—to develop useful tools and contribute to the field of Islamic studies.

7 Acknowledgments

We thank all participating teams for their contributions, as well as the organizers of the Arabic NLP 2025 Conference and the broader community for their support. We are grateful to our annotation team for ensuring the authenticity and appropriateness of our resources. This shared task was funded by the ARG grant ARG01-0524-230318, awarded by the Qatar Research, Development, and Innovation Council (QRDI). The efforts of Dr. Wajdi Zaghouani were partially supported by the NPRP14C-0916-210015 grant from QRDI.

References

Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur A. Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed K. Elmagarmid, Mohamed Y. Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, and 22 others. 2025. Fanar: An arabic-centric multimodal generative AI platform. *arXiv preprint*, arXiv:2501.13944.

Muhammad Abu Ahmad, Mohamad Ballout, Raia Abu Ahmad, and Elia Bruni. 2025. Transformer tafsir at qias 2025 shared task: Hybrid retrieval-augmented generation for islamic knowledge question answering. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alaa N Akkila and Samy S Abu Naser. 2016. Proposed expert system for calculating inheritance in islam. *International Journal of Advanced Research in Computer Science*.

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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydin. 2024. A rag-based question answering system proposal for understanding islam: Mufassirqas llm. *arXiv preprint arXiv:2401.15378*.

Nourah Alangari and Nouf AlShenaifi. 2025. N&n at qias 2025: Chain-of-thought ensembles with retrieval-augmented framework for classical arabic islamic. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Nouar AlDahoul 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Hayfa A Aleid and Aqil M Azmi. 2025. Hajj-fqa: A benchmark arabic dataset for developing question-answering systems on hajj fatwas: H. aleid and a. azmi. *Journal of King Saud University Computer and Information Sciences*, 37(6):135.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, and 1 others. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Is gpt-4 a good islamic expert for answering quran questions? In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 124–133.

Sanaa Alowaidi, Eric Atwell, and Mohammed Ammar Alsalka. 2025. Sea-team at qias 2025: Enhancing llms for question answering in islamic texts. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Shatha Altammami. 2025. Sha at the qias shared task: Llms for arabic islamic inheritance reasoning. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

M. Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan Alrashed, Faisal Mirza, Shaykhah Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, and 6 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Salah Eddine Bekhouche, Abdellah Zakaria Sellam, Hichem Telli, Cosimo Distante, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Abdessalam Bouchekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al-Khatib, and Mohammed Ghaly. 2025. Assessing large language models on islamic legal reasoning: Evidence from inheritance law evaluation. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 43 others. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*. Rapport technique, Équipe Gemini, Google.

Eman Elrefai, Abdelrahman Ahmad, Aml Hassan Esmail, and Mohamed Lotfy Elrefai. 2025. Gumball at qias 2025: Arabic llm automated reasoning in islamic inheritance. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, 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, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Juncai He, and 1 others. 2023. Acegpt, localizing large language models in arabic. *arXiv preprint arXiv:2309.12053*.

Anis Koubaa, Adel Ammar, Lahouari Ghouti, Omar Najar, and Serry Sibaee. 2024. Arabiangpt: Native arabic gpt-based large language model. *arXiv preprint arXiv:2402.15313*.

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*, pages 79–87, Marseille, France. European Language Resources Association.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an qa 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore.

Marryam Yahya Mohammed, Sama Ayman Ali, Salma Khaled Ali, Ayad Abdul Majeed, and Ensaf Hussein Mohamed. 2025. Aftina: enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.

Yossra Noureldien, Hassan Suliman, Farah Attallah, Abdelrazig Mohamed, and Sara Abdalla. 2025. Athar at qias2025: Mcqs-based question answering systems for islamic inheritance and classical knowledge. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Faiza Qamar, Seemab Latif, and Rabia Latif. 2024. A benchmark dataset with larger context for non-factoid question answering over islamic text. *arXiv preprint arXiv:2409.09844*.

Jihad R'baiti, Chouaib El Hachimi, Youssef Hmamouche, and Amal 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Muhammad Razif Rizqullah, Ayu Purwarianti, and Alham Fikri Aji. 2023. Qasina: Religious domain question answering using sirah nabawiyah. In *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, pages 1–6.

Mohamed Samy, Mayar Boghdady, Marwan El Adawi, Mohamed Nassar, and Ensaf Hussein. 2025. Tokenizers united at qias 2025: Rag-enhanced question answering for islamic studies by integrating semantic retrieval with generative reasoning. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Mohammad Amaan Sayeed, Mohammed Talha Alam, Raza Imam, Shahab Saquib Sohail, and Amir Hussain. 2025. From rag to agentic: Validating islamic-medicine responses with llm agents. *arXiv preprint arXiv:2506.15911*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Serry Sibaee, Mahmoud Reda, Omer Nacar, Yasser Alhabashi, Adel Ammar, and Wadii Boulila. 2025. Anlpers at qias 2025: Cot for islamic inheritance. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Sadia Tabassum, AHM Hoque, Sharaban Twahura, and Mohammad Osiur Rahman. 2019. Developing an islamic farayez system applying software engineering. *Jurnal Kejuruteraan*, 31(1):25–38.

Osama Zaki, Asmaa Badawy, Nada Elgewily, and Ahmed Sharaf. 2025. Cis-rg at qias 2025: Assessing large language models on islamic legal reasoning and mathematical calculations. In *Proceedings of The Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University-Computer and Information Sciences*, 33(1):68–76.

NYUAD at QIAS Shared Task: Benchmarking the Legal Reasoning of LLMs in Arabic Islamic Inheritance Cases

Nouar AlDahoul
Computer Science Department
New York University
Abu Dhabi, UAE
nouar.aldahoul@nyu.edu

Yasir Zaki
Computer Science Department
New York University
Abu Dhabi, UAE
yasir.zaki@nyu.edu

Abstract

Islamic inheritance domain holds significant importance for Muslims to ensure fair distribution of shares between heirs. Manual calculation of shares under numerous scenarios is complex, time-consuming, and error-prone. Recent advancements in Large Language Models (LLMs) have sparked interest in their potential to assist with complex legal reasoning tasks. This study evaluates the reasoning capabilities of state-of-the-art LLMs to interpret and apply Islamic inheritance laws. We utilized the dataset proposed in the ArabicNLP QIAS 2025 challenge, which includes inheritance case scenarios given in Arabic and derived from Islamic legal sources. Various base and fine-tuned models, are assessed on their ability to accurately identify heirs, compute shares, and justify their reasoning in alignment with Islamic legal principles. Our analysis reveals that the proposed majority voting solution, leveraging three base models (Gemini Flash 2.5, Gemini Pro 2.5, and GPT o3), outperforms all other models that we utilized across every difficulty level. It achieves up to 92.7% accuracy and secures third place overall in the challenge[1] (Bouchekif et al., 2025a).

1 Introduction

Islamic inheritance, which is known as “Ilm al-Mawārīth” in Arabic, is an area of jurisprudence that is highly structured, rule-based, and sensitive to context (Bouchekif et al., 2025a,b). The Qur’an introduced various rights and restrictions related to inheritance, marking significant improvements in the treatment of women and family relations for its time[2]. It also aimed to establish clear and fixed inheritance laws, contributing to the formation of a comprehensive legal system.

Islamic inheritance jurisprudence aims to prevent disputes by clearly defining the shares of each heir[2]. It ensures fair and equitable distribution, though Qur’anic verses assign different shares to specific relatives. Inheritance domain holds significant importance for Muslims, as it determines the rightful heirs, the individuals to be inherited from, and the specific shares allocated to each heir (Zouaoui and Rezeg, 2021). Upon a person’s death, a matter of particular concern is the management of all the property left behind. Manual calculation is a complex, time-consuming, and error-prone task that can be extremely difficult and costly. Automation of this calculation is convenient to save time, effort, and cost (Zouaoui and Rezeg, 2021).

Our analyses and experiments center around the following research questions: **RQ1**: Do current Arabic open-source LLMs perform well in Islamic inheritance reasoning? **RQ2**: To what extent do state-of-the-art proprietary base LLMs excel in Islamic inheritance reasoning? and **RQ3**: Can fine-tuning LLMs for the inheritance reasoning task improve performance?

We address **RQ1** by running several open-source Arabic LLMs. To answer **RQ2**, we utilized APIs of state-of-the-art proprietary LLMs. Additionally, to answer **RQ3**, we fine-tuned several LLMs with the inheritance multiple-choice questions dataset.

2 Related Work

LLMs have shown impressive performance in a variety of natural language understanding tasks (AlDahoul et al., 2024a; Kuo et al., 2025; AlDahoul et al., 2024b). When it comes to representing Islam, it is essential to ensure that its beliefs and teachings are portrayed accurately and faithfully, grounded in the Quran and Sunnah (Patel et al., 2023). Additionally, it is important to prevent hallucination in Islamic fatwa generation (Mohammed et al., 2025).

Several studies have focused on automating the inheritance calculation. (Jimoh et al., 2014) used an

[1] https://sites.google.com/view/qias2025/leaderboards?authuser=0

[2] https://islamicwillstrust.com/islamic-law-of-inheritance/

expert system to calculate shares based on Islamic law. Despite the growing development of automated knowledge retrieval systems, few leverage semantic web technologies for Islamic knowledge, particularly in Arabic. (Zouaoui and Rezeg, 2021) introduced AraFamOnto, an Arabic ontology-based system designed to automate Islamic inheritance calculations by efficiently modeling family relationships and reducing manual effort.

One work[3] examined the capabilities of several generative AI models in applying the principles of Islamic inheritance law. In their experiment, although ChatGPT-4[4] surpasses other models in performance, it continues to exhibit notable limitations, including fabricated references and legal inaccuracies. There is an ongoing effort to transform Islamic studies through generative AI as part of the 2024–2027 project[5]. This initiative focuses on building AI tools to engage with classical and contemporary Islamic texts.

However, previous efforts to automate inheritance problem solving using generative models remain limited, both in the number of models explored and due to the absence of large-scale datasets encompassing diverse scenarios and difficulty levels. This study addresses these gaps by utilizing a comprehensive dataset (Bouchekif et al., 2025a,b) of inheritance cases to evaluate the performance of state-of-the-art LLMs.

3 Materials and Methods

3.1 Dataset Overview

The QIAS (Question-and-Answer in Islamic Studies Assessment Shared Task) 2025 (Bouchekif et al., 2025a,b) dataset for Islamic inheritance reasoning contains multiple-choice questions (MCQs) categorized into three difficulty levels: beginner, intermediate, and advanced. The primary goal is to evaluate the reasoning abilities of LLMs within the domain of Islamic knowledge. Each question has exactly one correct answer and presents six answer choices, labeled A through F, each accompanied by a corresponding textual explanation. The dataset is annotated using one of six distinct symbols. The questions are sourced from a corpus of 32,000 fatwas from IslamWeb and have been validated by a qualified expert in Islamic inheritance law. The dataset is divided into 9,446 examples for training, 1,000 examples for validation, and 1,000 examples for testing in the final test phase.

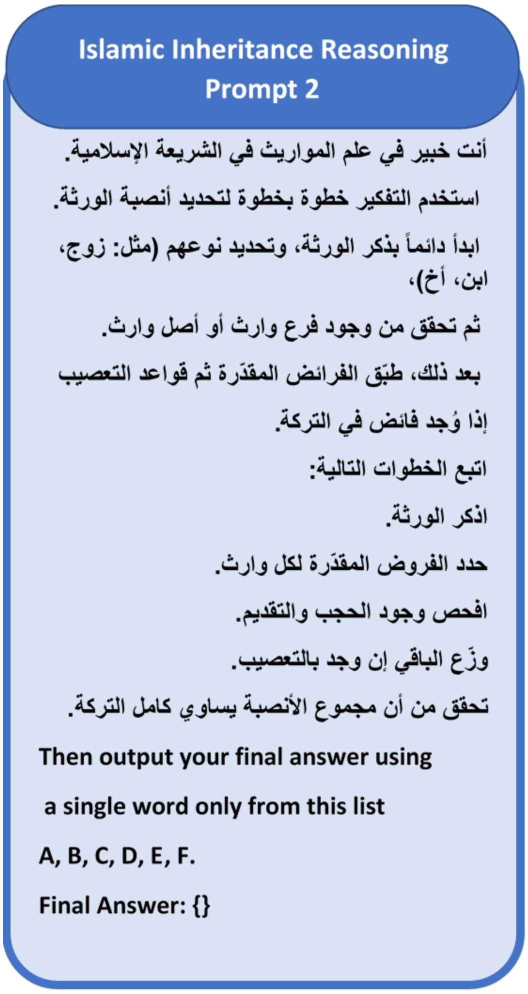
3.2 Methods

We have evaluated several methods, including base and fine-tuned LLMs, to find the best solution for Islamic inheritance reasoning. We utilized two prompts: Prompt 1 and Prompt 2. While Prompt 1 has simple structure, Prompt 2 has zero-shot Chain-of-Thought (CoT) style for solving MCQs by thinking step-by-step and justifying reasoning clearly.

Prompt 1: Islamic inheritance reasoning

Answer the following question using a single word only from this list: A, B, C, D, E, F.
Final Answer:

In the first experiment, base open-source Arabic models such as Falcon3 (Almazrouei



Islamic Inheritance Reasoning
Prompt 2

أنت خبير في علم المواريث في الشريعة الإسلامية.

استخدم التفكير خطوة بخطوة لتحديد أنصبة الورثة.

ابدأ دائماً بذكر الورثة، وتحديد نوعهم (مثل: زوج، ابن، أخ)،

ثم تحقق من وجود فرع وارث أو أصل وارث.

بعد ذلك، طبّق الفرائض المقدّرة ثم قواعد التعصيب

إذا وُجد فائض في التركة.

اتبع الخطوات التالية:

اذكر الورثة.

حدد الفروض المقدّرة لكل وارث.

افحص وجود الحجب والتقديم.

وزّع الباقي إن وجد بالتعصيب.

تحقق من أن مجموع الأنصبة يساوي كامل التركة.

Then output your final answer using

a single word only from this list

A, B, C, D, E, F.

Final Answer: {}

[3] https://islamiclaw.blog/2025/04/03/roundtable-augmented-learning-generative-artificial-intelligence-and-islamic-inheritance-law/

[4] https://openai.com/index/chatgpt/

[5] https://www.cilecenter.org/research-publications/funded-projects/transforming-islamic-studies-age-generative-artificial

et al., 2023) ("tiiuae/Falcon3-7B-Instruct")[6],[7], Fanar ("QCRI/Fanar-1-9B-Instruct")[8] (Team et al., 2025), and Allam ("ALLaM-AI/ALLaM-7B-Instruct-preview")[9] (Bari et al., 2024) were assessed. Additionally, we utilized "Allam thinking" ("almaghrabima/ALLaM-Thinking")[10] , an advanced Arabic LLM, which was fine-tuned and optimized specifically for reasoning and math. It was also prompted to think step-by-step.

Additionally, proprietary models such as Gemini Flash 2.5, Gemini Pro 2.5[11] (Team et al., 2023), GPT-4o (Hurst et al., 2024), and GPT o3[12], were evaluated for the Islamic inheritance reasoning task using the APIs of their base models. All previous LLMs were assessed in inference mode using Prompt 2 with temperature set to 0 and top_p set to 1.

To improve the performance of the LLMs in reasoning and increase the rate of correct answers to inheritance questions, we fine-tuned several open-source and proprietary LLMs such as GPT-4o, Gemini Flash 2.5, and Llama 4 Scout[13], [14] ("meta-llama/Llama-4-Scout-17B-16E-Instruct"). All LLMs were fine-tuned in a supervised learning setting, with a training set of 7,000 examples and a validation set of 2,446 examples used during training. The results of the comparison between all LLMs, including the base and fine-tuned, are reported using the 1,000 examples in the validation set.

Llama 4 was fine-tuned utilizing two prompts: Prompt 1 and Prompt 2. The fine-tuning was done using Low-Rank Adaptation (LoRA) (Hu et al., 2022) as the Parameter-Efficient Fine-Tuning (PEFT) method. The training was carried out for seven epochs with a learning rate of 0.0002. Both the training and evaluation batch sizes were set to 1 per device, and the gradient accumulation steps were set to 1. The optimizer used was paged_adamw_32bit. Additionally, 10 warmup steps were used to stabilize the initial training phase. We loaded the model using 4-bit quantization for memory efficiency. The fine-tuned models have been uploaded to Hugging Face: `https://huggingface.co/NYUAD-ComNets/NYUAD_Llama4_Inheritance_Solver`, and `https://huggingface.co/NYUAD-ComNets/NYUAD_Llama4_Inheritance_Solver2`.

GPT-4o was fine-tuned in two scenarios: without a system prompt and with system Prompt 2. The fine-tuning was done on the OpenAI platform for 5 epochs, with a learning rate multiplier of 2 and automatically selected batch size. Similarly, Gemini Flash 2 and 2.5 were fine-tuned with system Prompt 2. The fine-tuning was done on the Google AI Vertex platform. The hyper-parameters used for fine-tuning are 3 epochs and a learning_rate_multiplier of 5. Flash 2.5 used an adapter size of 1, while Flash 2 used an adapter size of 8 to train more parameters. In the inference phase, thinking was enabled.

3.3 Results and Discussion

Table 1 shows the accuracy of base LLMs for several open-source Arabic LLMs and proprietary state-of-the-art LLMs. Among the open models, Allam demonstrates relatively better performance (38.8%), indicating it may be more effectively tuned for this specific task or domain. In contrast, Falcon3 and Fanar perform worse, likely due to limited domain understanding. Although "Allam Thinking" was optimized for reasoning and math, its accuracy declined compared to the base Allam model. This set of experiments indicates a lack of domain knowledge and reasoning in the base open-source Arabic LLMs, which addresses **RQ1**.

Additionally, both GPT o3 (92.3%) and Gemini Flash 2.5 (91.5%) demonstrate the strongest performance, highlighting their advanced capability in understanding and reasoning about Islamic inheritance, which answers **RQ2**. The small gap in accuracy may be due to the results being based on a single run. Therefore, for future work, we should run each model multiple times and compute the average and variance. This process can provide a clearer comparison between the two.

Table 2 shows the prompt sensitivity of each

[6] `https://huggingface.co/blog/falcon3`

[7] `https://huggingface.co/tiiuae/Falcon3-7B-Instruct`

[8] `https://huggingface.co/QCRI/Fanar-1-9B-Instruct`

[9] `https://huggingface.co/ALLaM-AI/ALLaM-7B-Instruct-preview`

[10] `https://huggingface.co/almaghrabima/ALLaM-Thinking`

[11] `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#enhanced-reasoning`

[12] `https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf`

[13] `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`

[14] `https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E-Instruct`

LLM	Accuracy (%)
Falcon3	24.2
Fanar	31.7
Allam	38.8
Allam think	29.2
Gemini Flash 2.5	**91.5**
GPT-4o	70.1
GPT O3	**92.3**

Table 1: Accuracy of different Base LLMs.

LLM using Prompt 1 and Prompt 2. The GPT-4o model demonstrates a notable sensitivity to prompt design. When evaluated with Prompt 1, its accuracy is relatively low at 57.5%. However, a shift to Prompt 2 significantly enhances the performance to 70.1%, thereby. Allam also shows the same trend. On the other hand, Gemini Flash 2.5 exhibits high accuracy regardless of prompt content, suggesting strong internal reasoning and understanding.

LLM	Prompt	Accuracy (%)
Allam think	Prompt 1	28.8
Allam think	Prompt 2	**29.2**
Allam	Prompt 1	30.4
Allam	Prompt 2	**38.8**
Fanar	Prompt 1	28.7
Fanar	Prompt 2	**31.7**
Falcon	Prompt 1	**24.2**
Falcon	Prompt 2	22.8
Gemini Flash 2.5	Prompt 1	90.7
Gemini Flash 2.5	Prompt 2	**91.5**
GPT-4o	Prompt 1	57.5
GPT-4o	Prompt 2	**70.1**

Table 2: Prompt sensitivity: Accuracy of LLMs using two different Prompts.

When GPT-4o was fine-tuned for reasoning using the training set, its accuracy improved significantly, reaching over 84% as shown in Table 3. On the other hand, when no system prompt is used for tuning, the accuracy reaches 84.7%. When we added "Prompt 2" as a system prompt, the accuracy improved to 86.6%. On the contrary, the performance of Gemini Flash 2.5 dropped after fine-tuning (91.5% –> 74.6%) on this task. This disparity in the fine-tuning results between GPT-4o and Gemini Flash 2.5 gives an answer to **RQ3**.

The reason behind this disparity in performance after fine-tuning is that GPT-4o is a highly generalist model, not specialized for reasoning. This makes it more adaptable to niche domains like Islamic inheritance when given domain-specific data. Furthermore, GPT-4o may have moderate prior knowledge about Islamic inheritance laws, so fine-tuning filled a knowledge gap rather than conflicting with existing knowledge.

On the other hand, we observed performance degradation in fine-tuning Gemini Flash 2.5 with the adapter size set to 1 (tuning fewer parameters). To confirm our observation, we may consider fine-tuning the same model with larger adapter sizes. The reason for degradation may stem from the fact that Flash 2.5 is optimized for CoT reasoning. If the fine-tuning dataset has only final labels and lacks detailed reasoning chains, the model may lose its reasoning structure. This results in misalignment between what it's trained to do and what it's fine-tuned to.

Fine-tuning Flash 2 with an adapter size of 8 resulted in degraded performance on this task. This may be due to the relatively limited size of the fine-tuning dataset, which was insufficient to train a larger adapter. As a result, the model likely failed to generalize well. To validate our observation, we may consider fine-tuning the same model with smaller adapter sizes or using the base model.

As shown in Table 3, fine-tuning GPT-4o with Prompt 2 resulted in a better performance, which contrasts with the behavior of Llama 4 Scout.

LLM	Prompt	Accu. (%)
Fine-tuned Llama4 Scout	Prompt 1	84.3
Fine-tuned Llama4 Scout	Prompt 2	82.4
Fine-tuned Gemini Flash 2	Prompt 2	64.6
Fine-tuned Gemini Flash 2.5	Prompt 2	74.6
Fine-tuned GPT-4o	No system Prompt	84.7
Fine-tuned GPT-4o	Prompt 2	86.6

Table 3: Accuracy of fine-tuned LLMs.

Finally, we ran an experiment to assess the best-performing base LLMs with the testing set released in the test phase and included a set of 1000 MCQs. We ran three base models in the inference mode. The accuracies of GPT-o3, Gemini Flash 2.5, and Gemini Pro 2.5 on the testing data were 88.4%, 88.1%, and 87.9%, respectively. We later applied a majority voting technique using the predictions from these three LLMs, resulting in a final accuracy of 92.7%, which secured third place overall in the challenge. Figure 1 presents MCQ example labeled with the correct answer 'D'. However, both GPT O3 in Figure 2 and Gemini Pro 2.5 in Figure 3 selected option 'E' and provided step-by-step reasoning to justify their choice.

Limitations

The limitation of this work is that LLMs often lack comprehensive knowledge of all inheritance scenarios, which leads even the best reasoning models to occasionally select incorrect answers. Fine-tuning can help address this issue. However, a new problem arises: in the original dataset, answer choices were labeled without detailed reasoning for each option, which complicated the fine-tuning of reasoning-based models. To solve this, a second version of the dataset was released recently, including the reasoning behind each selected answer. This may show potential for increasing the accuracy.

References

Nouar AlDahoul, Talal Rahwan, and Yasir Zaki. 2024a. Polytc: a novel bert-based classifier to detect political leaning of youtube videos based on their titles. *Journal of Big Data*, 11(1):80.

Nouar AlDahoul, Myles Joshua Toledo Tan, Harishwar Reddy Kasireddy, and Yasir Zaki. 2024b. Advancing content moderation: Evaluating large language models for detecting sensitive content across text, images, and videos. *arXiv preprint arXiv:2411.17123*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan AlRashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, et al. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

RG Jimoh, IR Adebayo, AO Ameen, and HO Ahmad. 2014. Design of an expert system to calculate inheritance shares based on islamic law. *IEEE-African Journal of Computing & ICTs*, 7(2):151–156.

Chen Wei Kuo, Kevin Chu, Nouar AlDahoul, Hazem Ibrahim, Talal Rahwan, and Yasir Zaki. 2025. Neutralizing the narrative: Ai-powered debiasing of online news articles. *arXiv preprint arXiv:2504.03520*.

Marryam Yahya Mohammed, Sama Ayman Ali, Salma Khaled Ali, Ayad Abdul Majeed, and Ensaf Hussein Mohamed. 2025. Aftina: enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.

Shabaz Patel, Hassan Kane, and Rayhan Patel. 2023. Building domain-specific llms faithful to the islamic worldview: Mirage or technical possibility? *arXiv preprint arXiv:2312.06652*.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, et al. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

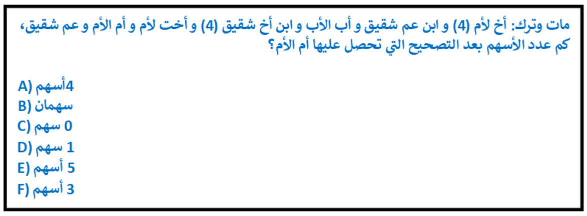
Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University-Computer and Information Sciences*, 33(1):68–76.

A Appendix

A.1 Reasoning Analysis

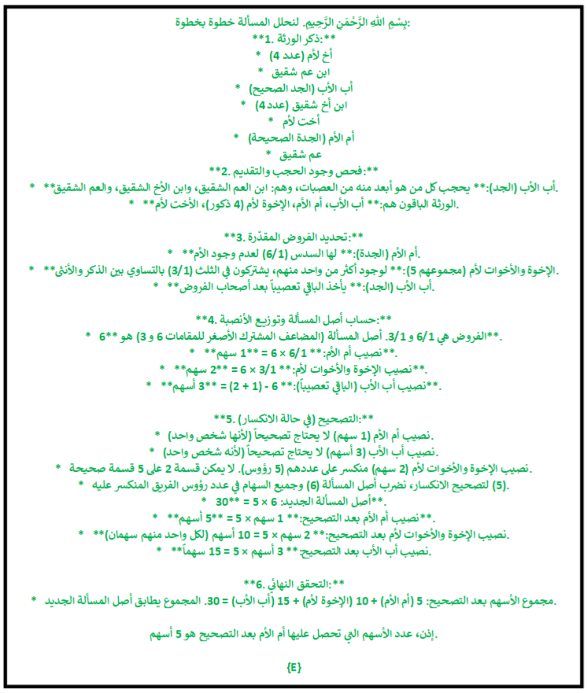
Figure 1 presents MCQ example labeled with the correct answer 'D'. However, both GPT O3 in Figure 2 and Gemini Pro 2.5 in Figure 3 selected option 'E' and provided step-by-step reasoning to justify their choice.



مات وترك: أخ لأم (4) و ابن عم شقيق و أب الأب و ابن أخ شقيق (4) و أخت لأم و أم الأم و عم شقيق،
كم عدد الأسهم بعد التصحيح التي تحصل عليها أم الأم؟

A) 4أسهم
B) سهمان
C) 0 سهم
D) 1 سهم
E) 5 أسهم
F) 3 أسهم

Figure 1: MCQ Example



بِسْمِ اللهِ الرَّحْمَنِ الرَّحِيمِ. لنحلل المسألة خطوة بخطوة:
1. ذكر الورثة:
* أخ لأم (عدد 4)
* ابن عم شقيق
* أب الأب (الجد الصحيح)
* ابن أخ شقيق (عدد 4)
* أخت لأم
* أم الأم (الجدة الصحيحة)
* عم شقيق
2. فحص وجود الحجب والتقديم:
* .أب الأب (الجد):** يحجب كل من هو أبعد منه من العصبات، وهم: ابن العم الشقيق، وابن الأخ الشقيق، والعم الشقيق**
* .الورثة الباقون هم:** أب الأب، أم الأم، الإخوة لأم (4 ذكور)، الأخت لأم**

3. تحديد الفروض المقدّرة:
* .أم الأم (الجدة):** لها السدس (6/1) لعدم وجود الأم**
* .الإخوة والأخوات لأم (مجموعهم 5):** لوجود أكثر من واحد منهم، يشتركون في الثلث (3/1) بالتساوي بين الذكر والأنثى**
* .أب الأب (الجد):** يأخذ الباقي تعصيباً بعد أصحاب الفروض**

4. حساب أصل المسألة وتوزيع الأنصبة:
* .**الفروض هي 6/1 و 3/1. أصل المسألة (المضاعف المشترك الأصغر للمقامات 6 و 3) هو **6
* .**نصيب أم الأم:** 6/1 × 6 = **1 سهم**
* .**نصيب الإخوة والأخوات لأم:** 3/1 × 6 = **2 سهم**
* .**نصيب أب الأب (الباقي تعصيباً):** 6 - (1 + 2) = **3 أسهم**

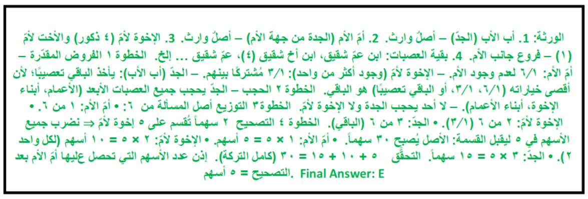
5. التصحيح (في حالة الانكسار):
* .نصيب أم الأم (1 سهم) لا يحتاج تصحيحاً (لأنها شخص واحد)
* .نصيب أب الأب (3 أسهم) لا يحتاج تصحيحاً (لأنه شخص واحد)
* .نصيب الإخوة والأخوات لأم (2 سهم) منكسر على عددهم (5 رؤوس). لا يمكن قسمة 2 على 5 قسمة صحيحة
* .(5) لتصحيح الانكسار، نضرب أصل المسألة (6) وجميع السهام في عدد رؤوس الفريق المنكسر عليه
* .**أصل المسألة الجديد: 6 × 5 = **30
* .**نصيب أم الأم بعد التصحيح:** 1 سهم × 5 = **5 أسهم**
* .نصيب الإخوة والأخوات لأم بعد التصحيح:** 2 سهم × 5 = 10 أسهم (لكل واحد منهم سهمان)**
* .نصيب أب الأب بعد التصحيح:** 3 أسهم × 5 = 15 سهماً**

6. التحقق النهائي:
* .مجموع الأسهم بعد التصحيح: 5 (أم الأم) + 10 (الإخوة لأم) + 15 (أب الأب) = 30. المجموع يطابق أصل المسألة الجديد

.إذن، عدد الأسهم التي تحصل عليها أم الأم بعد التصحيح هو 5 أسهم

{E}

Figure 2: Generated output from Gemini Pro 2.5 for MCQ Example



الورثة: 1. أب الأب (الجدّ) – أصلٌ وارث. 2. أمّ الأم (الجدة من جهة الأم) – أصلٌ وارث. 3. الإخوة لأمّ (٤ ذكور) والأخت لأمّ
(١) – فروع جانب الأم. 4. بقية العصبات: ابن عمّ شقيق، ابن أخ شقيق (٤)، عمّ شقيق … إلخ. الخطوة ١ الفروض المقدّرة –
أمّ الأم: ٦/١ لعدم وجود الأم. – الإخوة لأمّ (وجود أكثر من واحد): ٣/١ مُشتركًا بينهم. – الجدّ (أب الأب): يأخذ الباقي تعصيبًا؛ لأن
أقصى خياراته (٦/١، ٣/١، أو الباقي تعصيبًا) هو الباقي. الخطوة ٢ الحجب – الجدّ يحجب جميع العصبات الأبعد (الأعمام، أبناء
الإخوة، أبناء الأعمام). – لا أحد يحجب الجدة ولا الإخوة لأمّ. الخطوة ٣ التوزيع أصل المسألة من ٦: • أمّ الأم: ١ من ٦. •
الإخوة لأمّ: ٢ من ٦ (٣/١). • الجدّ: ٣ من ٦ (الباقي). الخطوة ٤ التصحيح ٢ سهماً تُقسم على ٥ إخوة لأمّ ⇒ نضرب جميع
الأسهم في ٥ ليقبل القسمة: الأصل يُصبح ٣٠ سهماً. • أمّ الأم: ١ × ٥ = ٥ أسهم. • الإخوة لأمّ: ٢ × ٥ = ١٠ أسهم (لكل واحد
٢). • الجدّ: ٣ × ٥ = ١٥ سهماً. التحقُّق ٥ + ١٠ + ١٥ = ٣٠ (كامل التركة). إذن عدد الأسهم التي تحصل عليها أمّ الأم بعد
التصحيح = ٥ أسهم. Final Answer: E

Figure 3: Generated output from GPT O3 for MCQ Example

SHA at the QIAS Shared Task: LLMs for Arabic Islamic Inheritance Reasoning

Shatha Altammami
King Saud University, Saudi Arabia
shaltammami@ksu.edu.sa

Abstract

This paper presents our system for SubTask 1: Islamic Inheritance Reasoning in the QIAS 2025 Shared Task, which evaluates large language models (LLMs) on (*ilm al-mawārīth*) (Islamic science of inheritance) using a benchmark of Arabic multiple-choice questions (MCQs) derived from expert-reviewed fatwas. We explore static and dynamic few-shot prompting, retrieval-augmented generation (RAG) with a large fatwa corpus, and a progressive n-gram overlap retrieval method. The n-gram method is applied both to select the top five most similar MCQs for dynamic prompting and to retrieve the most relevant fatwa answer as additional context during inference. We evaluate proprietary and open-source LLMs individually and in ensemble form. Results show that dynamic prompting and RAG consistently improve accuracy across our best performing model, Gemini, achieving 62.26% accuracy on the test set.

1 Introduction

Large Language Models (LLMs) have achieved impressive advances in reasoning and problem solving(Plaat et al., 2024; Zhao et al., 2023), yet their performance often varies across languages and domains (Matarazzo and Torlone, 2025). While most prior work has focused on English, a growing body of research has examined Arabic, revealing mixed results in comprehension and complex reasoning (Khondaker et al., 2024).

One underexplored domain is (*ilm al-mawārīth*) (Islamic science of inheritance), which requires mapping textual descriptions of heirs to precise share distributions — a task demanding multi-step reasoning and domain-specific accuracy. To enable systematic evaluation in this underexplored domain, the QIAS 2025 Shared Task offers a large-scale benchmark of Arabic multiple-choice questions (MCQs) on (*ilm al-mawārīth*) (Bouchekif et al., 2025a,b).

In this paper, we describe our system for the shared task, which integrates static and dynamic few-shot prompting, retrieval-augmented generation (RAG) using a large fatwa corpus, and a progressive n-gram overlap retrieval method. The n-gram method is employed in two ways: (1) to retrieve the top five most similar MCQs from the training set for dynamic prompting, and (2) to identify the most relevant fatwa question and extract its answer as contextual input for inference.

We evaluate both proprietary and open-source models, individually and in an ensemble configuration. Results show that dynamic prompting and RAG provide consistent improvements, with our best-performing model, Gemini, achieving 62.26% accuracy on the test set.

2 Related Work

LLMs have demonstrated strong performance on text-based multiple-choice questions (MCQs), particularly in factual recall and reading comprehension tasks(Matarazzo and Torlone, 2025). Proprietary models such as GPT-4 and Gemini consistently achieve high accuracy on knowledge-based and standardized exam questions, with documented success on domains such as the Dental Admission Test (DAT) (Hou et al., 2025). Similarly, large open-source models like LLaMA3-70B perform competitively in natural sciences and reading comprehension domains (Hou et al., 2025). However, these models consistently struggle with higher-order cognitive skills, multi-step reasoning, and advanced mathematical problem solving, with hallucination remaining a persistent issue, especially in complex reasoning scenarios (Saxena et al., 2024).

Although most prior work has focused on evaluating models in English, some studies have examined Arabic. Existing research shows that LLMs demonstrate mixed performance in comprehend-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 867–872
November 8-9, 2025 ©2025 Association for Computational Linguistics

ing Arabic content and solving complex reasoning tasks. Proprietary models such as GPT-4 and GPT-3.5 perform competitively but are often outperformed by smaller, fine-tuned Arabic models on domain-specific tasks (Khondaker et al., 2023). In contrast, open-source models like LLaMA-3-70B still lag behind both ChatGPT and specialized Arabic models, partly due to limited Arabic representation in large pretraining corpora and high sensitivity to input phrasing (Khondaker et al., 2024).

The closest relevant evaluation is the Qur'an Question Answering shared task (Malhas et al., 2022, 2023), which addressed Machine Reading Comprehension (MRC) over Classical Arabic text. It highlighted the Qur'an's linguistic complexity and topic diversity. Their results emphasize the gap between general Arabic NLP progress and the sensitive religious domains..

Despite this growing body of work, there is a clear research gap: no empirical studies have systematically evaluated LLMs' accuracy in answering questions across diverse areas of Islamic scholarship. Current literature focuses on general NLP benchmarks and professional examinations, leaving domain-specific tasks such as Islamic jurisprudence (fiqh) and inheritance law (*ilm al-mawarith*) largely unexplored.

3 Task Description

The shared task focuses on evaluating LLMs in the Islamic domain, with a particular emphasis on their ability to reason about inheritance-related scenarios (*ilm al-mawārīth*). In this subtask, each multiple-choice question (MCQ) presents a specific inheritance case describing a set of heirs, and the proposed model must determine the correct distribution outcome by selecting the right option from a predefined set of answers. The evaluation is based on classification accuracy over a held-out test set, ensuring an objective comparison of model performance.

4 Dataset

Experiments were conducted using the official dataset provided for the Islamic Inheritance Reasoning task. The dataset comprises multiple-choice questions (MCQs) drawn from authentic Islamic jurisprudential sources and are designed to test not only factual recall but also the model's ability to apply complex, rule-based reasoning grounded in Islamic law. Also a supplementary fatwa corpus is provided which we used for the retrieval-augmented generation (RAG)-based inference.

MCQ Dataset

- **Training Set:** 20,000 MCQs, distributed across three difficulty levels: 500 Beginner, 300 Intermediate, and 200 Advanced.
- **Validation Set:** 1,000 MCQs, distributed across three difficulty levels: 500 Beginner, 300 Intermediate, and 200 Advanced.
- **Test Set:** 1,000 MCQs with hidden labels, balanced between 500 Beginner and 500 Advanced questions.

Each MCQ includes 4 to 6 answer options (A–F), with exactly one correct label. The questions span a wide range of inheritance scenarios requiring precise application of Islamic legal principles (*ilm al-mawārīth*).

Fatwa Corpus

In addition to the MCQ dataset, we used a corpus of 3,165 fatwas from IslamWeb to support retrieval-augmented generation (RAG). Stored as JSON files, each fatwa contains a user-submitted question and an expert legal response, offering rich, domain-specific context to enhance model reasoning.

5 Methodology

As baseline models, we fine-tuned the top-performing model from the Qur'an Question Answering shared task (Malhas et al., 2022, 2023), AraBERTv2 (Antoun et al., 2020), on the 20,000-question training set, achieving an accuracy of 47.4% on the validation (development) set. Another baseline(code was provided by the shared task organizers) involved prompting the Fanar LLM with two few-shot MCQ examples, which yielded 49.7% accuracy on the same validation (development) set. Building on the best-performing baseline, we consulted the literature and identified key areas for improvement.

5.1 Few-Shot In-Context Learning

Few-shot prompting has proven effective in eliciting structured reasoning from LLMs (Brown et al., 2020; Kojima et al., 2022). Static few-shot examples provide a general template for reasoning, whereas dynamic example selection can improve performance by aligning examples with the test instance (Liu et al., 2022). In this work, we explore both static and dynamic prompting strategies. In the static approach, five examples are included in the prompt for every question. In the dynamic approach, the five most similar questions are retrieved from the training set and included in the prompt. Similarity is determined using an n-gram overlap strategy previously introduced in Altammami et al. (2019), originally developed for segmenting and annotating Hadith corpora. The algorithm has been adapted for the current task, as explained in Section 5.3.

5.2 Retrieval-augmented generation

We utilized Retrieval-Augmented Generation (RAG), a widely recognized and effective approach for improving NLP tasks (Lewis et al., 2020; Wu et al., 2024), particularly in knowledge-intensive and domain-specific scenarios (Xiong et al., 2024). RAG consistently enhances answer accuracy, factuality, and adaptability compared to language models that rely solely on pre-trained knowledge (Siriwardhana et al., 2023).

Initial experiments using vector-based semantic similarity methods (e.g., FAISS) yielded suboptimal results. These approaches often failed to distinguish between conceptually distinct heirs (e.g., *son* vs. *daughter*), treating them as similar due to surface-level embedding similarities. This limitation is particularly problematic in the domain of Islamic inheritance law, where precise legal roles carry significant implications.

To address this, our system identifies the most similar fatwa question from a large corpus using the n-gram approach described in Section 5.3. The corresponding fatwa answer is then extracted and incorporated as additional context for the language model during inference.

5.3 Progressive n-gram Overlap

To support both dynamic few-shot prompting and retrieval-augmented generation (RAG), we developed a custom progressive n-gram overlap matching function. This method is used in two key places: (1) to select the most similar five MCQ questions from the training set for dynamic prompting, and (2) to identify the most relevant fatwa question from the Fatwa Corpus in order to extract its answer as additional context during inference.

Given a new inheritance question, the system iterates through the relevant dataset (training set or Fatwa Corpus) and compares the input question against all available questions using the progressive n-gram overlap function. The matching function follows a fallback strategy: It first computes trigram overlap, and if no sufficient match is found, it falls back to bigram or unigram overlap. We assign higher weights to longer n-grams to prioritize more specific matches: trigrams $w_3 = 1.0$, bigrams $w_2 = 0.5$, and unigrams $w_1 = 0.2$.

For dynamic prompting, the top five questions with the highest similarity scores from the training set are selected and included in the prompt. For RAG, the single highest-scoring fatwa question is selected from across all fatwa JSON files, and its Answer field is extracted and supplied to the language model as contextual input, as illustrated in Algorithm 1.

This approach ensures that retrieved examples and contextual fatwas are both semantically and structurally aligned with the input question, avoiding misleading matches that often occur with purely embedding-based similarity methods.

6 Experimental Design

6.1 Models

Four LLMs were evaluated in this study. Inference was configured to favor deterministic, short outputs by setting the temperature to 0.0 and the maximum output length to 2 tokens (sufficient to return a single uppercase letter).

- **Gemini-1.5-pro**: Google's generative language model, accessed via the Google Vertex AI API.
- **GPT-4**: OpenAI's GPT-4o model, accessed through the OpenAI API.
- **LLaMA**: Meta's LLaMA-3.3-70b model, accessed via the Groq API.
- **Fanar**: A domain-specific Arabic language model, accessed through a custom API.

6.2 Prompt Engineering

Three prompt engineering strategies were designed to evaluate their impact on model performance in Islamic inheritance reasoning:

- **Trial 1: Static Few-Shot In-Context Learning**
  A baseline configuration using five manually selected MCQ examples from the training

Model	Trial 1		Trial 2		Trial 3	
	Dev	Test	Dev	Test	Dev	Test
Gemini	60.50	57.30	60.10	61.40	61.80	**62.26**
GPT-4	58.30	55.55	57.60	47.90	54.10	46.30
Fanar	57.27	40.24	54.06	38.49	56.27	38.74
LLaMA	46.40	49.40	46.60	46.10	45.65	47.40
Ensemble	62.60	55.80	61.90	56.40	**63.40**	57.30

Table 1: Performance (%) of different models across three trials on the Islamic inheritance MCQ development and test datasets. Best results are shown in **bold**.

Algorithm 1: Progressive N-gram Matching for Fatwa Retrieval

```
Input   : Question $Q$;
          Set of Fatwa Files $\mathcal{F}$ (each
          containing Question, Answer fields)
Output : Best matching fatwa answer $A^*$
Initialize:
  $best\_score \leftarrow -\infty, \quad A^* \leftarrow$ None
foreach fatwa file $f \in \mathcal{F}$ do
    Load all questions $Q_f$ and answers $A_f$ from $f$;
    foreach candidate question $q \in Q_f$ do
        Normalize $Q$ and $q$ by removing punctuation, extra spaces;
        $score \leftarrow 0$;
        for $n \in \{3, 2, 1\}$ do
            Extract $n$-grams from $Q$ and $q$;
            Compute $overlap \leftarrow$ intersection of $n$-grams;
            Update $score \leftarrow score + w_n \times |overlap|$;
            Remove matched $n$-grams from further consideration;
        if $score > best\_score$ then
            $best\_score \leftarrow score$;
            $A^* \leftarrow$ corresponding answer to $q$;
return $A^*$
```

set. These examples were appended to each prompt uniformly, without regard to question similarity.

- **Trial 2: Dynamic Few-Shot In-Context Learning**
  Few-shot examples were dynamically selected for each input question using n-gram similarity from the training set. This ensured structural and semantic relevance between the input and the few-shot examples.

- **Trial 3: Dynamic Few-Shot In-Context Learning and RAG**
  In addition to dynamic example selection, the most similar fatwa question was retrieved using n-gram overlap, and the corresponding fatwa answer was appended to the prompt as context.

Model performance was assessed using accuracy of correctly answered 1,000 MCQs testing questions.

6.3 Results

Table 1 reports development and test accuracies across three independent trials. Gemini consistently outperforms other single models, achieving the highest test accuracy in Trial 3 (62.26%). GPT-4 performs competitively in Trial 1 but its accuracy declines sharply in later trials. Fanar and LLaMA lag behind, though LLaMA generally surpasses Fanar on the test set.

The ensemble method, based on majority voting, yields the best development accuracy in Trial 3 (63.40%) and consistently competitive results overall. Its improvements are more pronounced on development data than on test data, reflecting differences in dataset composition: The dev set contains beginner, intermediate, and advanced items, while the test set excludes intermediate items. This mismatch reduces the ensemble's generalization strength.

Gemini's steady gains suggest that it leverages additional retrieved context effectively, whereas GPT-4 appears more prone to "distraction," with the same context introducing noise and lowering accuracy. These contrasting behaviors highlight model-specific sensitivities to retrieval-augmented prompting, and further analysis is needed in future work to better understand how such distractions arise.

7 Conclusion

This paper presented our system for SubTask 1: Islamic Inheritance Reasoning in the QIAS

2025 Shared Task, where we evaluated static few-shot prompting, dynamic few-shot prompting, and dynamic prompting combined with retrieval-augmented generation, supported by a progressive n-gram overlap method. Evaluation on proprietary and open-source LLMs revealed that while some models experienced performance drops—suggesting that additional context can sometimes distract the model—others achieved consistent gains. Our best configuration (Gemini with RAG and dynamic prompting) reached 62.26% accuracy on the test set. Further analysis is required to better understand how retrieval context may distract certain models and how to design strategies that mitigate this effect.

Acknowledgments

I would like to thank the organizers and reviewers of this shared task.

References

Shatha Altammami, Eric Atwell, and Ammar Alsalka. 2019. Text segmentation using n-grams to annotate hadith corpus. In *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics*, pages 31–39.

Wissam Antoun and 1 others. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC*.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 2025a. Qias 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Second Arabic Natural Language Processing Conference, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 Second Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Yu Hou, Jay Patel, Liya Dai, Emily Zhang, Yang Liu, Zaifu Zhan, Pooja Gangwani, and Rui Zhang. 2025. Benchmarking of large language models for the dental admission test. *Health Data Science*, 5:0250.

Md Tawkat Islam Khondaker, Numaan Naeem, Fatimah Khan, Abdelrahim Elmadany, and Muhammad Abdul-Mageed. 2024. Benchmarking llama-3 on arabic language generation tasks. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 283–297.

Md. Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and M. Abdul-Mageed. 2023. Gptaraeval: A comprehensive evaluation of chatgpt on arabic nlp. *ArXiv*, abs/2305.14976.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. 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 *Proceedinsg 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.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *ArXiv*, abs/2501.04040.

A. Plaat, Annie Wong, Suzan Verberne, Joost Broekens, N. V. Stein, and T. Back. 2024. Reasoning with large language models, a survey. *ArXiv*, abs/2407.11511.

Yash Saxena, Sarthak Chopra, and Arunendra Mani Tripathi. 2024. Evaluating consistency and reasoning capabilities of large language models. In *2024 Second International Conference on Data Science and Information System (ICDSIS)*, pages 1–5. IEEE.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and 1 others. 2024. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6233–6251.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. A survey of large language models. *ArXiv*, abs/2303.18223.

ANLPers at QIAS: CoT for Islamic Inheritance

Serry Sibaee[1*] **Mahmoud Reda** [2] **Omer Nacar**[3] **Yasser Al-Habashi**[1]
Adel Ammar[1] **Wadii Boulila**[1]
[1]Prince Sultan University, Riyadh, Saudi Arabia
[2]Zagazig University
[3]Tuwaiq Academy – Tuwaiq Research and Development Center
{ssibaee, aammar, yalhabashi, wboulila}@psu.edu.sa
{20812017101263}@eng.zu.edu.eg
{o.najar}@tuwaiq.edu.sa
[*]Corresponding author: ssibaee@psu.edu.sa

Abstract

This paper presents a Chain-of-Thought (CoT) prompting approach for Islamic inheritance reasoning in multiple-choice question answering. We address the QIAS 2025 SubTask 1, which requires complex legal reasoning to determine correct inheritance shares according to Islamic jurisprudence. Our system employs two prompting strategies: direct answer extraction and step-by-step reasoning with regex-based answer extraction. We evaluate our approach using Claude 3.7 Sonnet and GPT-4o on Islamic inheritance MCQ tasks. Results demonstrate significant performance improvements when incorporating the thinking step: Claude 3.7 improved from 0.67 to 0.81, and GPT-4o from 0.63 to 0.74. Error analysis reveals that while models perform well in basic reasoning, they struggle with complex correction procedures (**Tasheeh**[1]) in inheritance calculations. Our findings confirm that structured reasoning substantially enhances LLM performance on complex Arabic legal reasoning tasks without requiring additional training or retrieval-augmented generation.

1 Introduction

The task of Islamic Inheritance Reasoning is a significant challenge in natural language processing and algorithmic systems due to the long process and mathematical operations needed to reach the final results. Current state-of-the-art models, such as those by (Sibaee et al., 2025), often struggle with answering a full hard inherence question such as: "Divide this inheritance according to Islamic law: The deceased left behind a father, a mother, the father's mother (paternal grandmother), a full brother, a full sister, a paternal half-brother, a paternal half-sister, a maternal half-brother, a maternal half-sister, and a nephew (son of full brother)." Sonnet-3.5 model did not show full logical thoughts. Sometimes, it reasoned correctly, but the final answer was wrong. This was also shown in (Abdulrahman and Walusimbi, 2024).

To address these limitations, we propose a new system based on showing the thought process of the model before answering the question. Our approach is novel because the type of problem is MCQ and this is an open field to show the logical thinking before answering. We hypothesize that this method will improve the performance obtained by (Wei et al., 2023). Our main contribution is showing the effectiveness of thinking and Chain of thoughts in answering hard and complex Islamic inheritance MCQ and this approach can help in using this SOTA models in these kind of tasks without adding any blocks to the pipeline e.g. RAG or finetuning.

2 Background

This study addresses QIAS 2025 – SubTask 1: *Islamic Inheritance Reasoning* (Bouchekif et al., 2025a), which focuses on answering multiple-choice questions (MCQs) related to the distribution of inheritance according to Islamic jurisprudence. The task inherently requires dual competencies: (1) comprehension of Arabic textual problem statements, and (2) the application of complex legal-mathematical reasoning to determine the precise share for each heir (Bouchekif et al., 2025b).

As an illustrative example, consider the scenario: *"A woman died leaving two sons, three daughters, a husband, a father, and a mother. What is the husband's share?"* with answer choices ranging from (A) Half to (F) Two-thirds. The correct answer, as prescribed by Islamic law, is (E) Quarter.

Previous research has explored various strategies for this domain. For instance, (Abdelazim et al., 2024) investigated the use of *Chain-of-*

[1] Tasheeh is the correction procedure applied when initial fractional shares of heirs do not divide evenly into whole numbers. It ensures integer shares while preserving proportional rights.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 873–877
November 8-9, 2025 ©2025 Association for Computational Linguistics

Thought (CoT) prompting to enhance performance in complex Arabic question-answering tasks, while (Zouaoui and Rezeg, 2021) employed ontology-based frameworks to model and solve Islamic inheritance problems, and (Sibaee et al., 2025) examined the LLMs on multiple topics including inheretince and showed a very low preformence in all of them.

Building upon these works, our approach integrates CoT reasoning to systematically decompose and solve inheritance problems step-by-step, followed by the application of regular expression (regex) techniques to accurately extract the final answer from the reasoning output. This combination is designed to address both the linguistic and jurisprudential complexity of the task, ensuring logically coherent reasoning and precise answer selection within the MCQ framework.

It is important to note that one of the most error-prone stages for existing models is the *Tasheeh* process, which requires adjusting fractional shares into integer values while maintaining proportional correctness. Our approach is designed to handle this step effectively within the CoT framework, thereby addressing a critical source of error in Islamic inheritance reasoning.

3 System Overview

Our system depends on calling an LLM to answer the given question in two ways: The first is to ask the question as is with the choices and ask the LLM to give the answer letter. The second method is to ask the model to explain the answer before selecting it, which is extracted using formatted regex.

4 Experimental Setup

We conducted the experiments using two SOTA LLMs. The first prompt is shown in Figure 1.

System Role: <You are a strict grader for multiple-choice exams.>

Task: <You must return only the correct choice letter (e.g., A, B, C...) without any explanation.>

Question: <question>

Options: <(options)>

Figure 1: LLM Prompt for Multiple Choice Answering

For the detailed (thinking) answer, the used prompt is displayed in Figure 2. The generic procedure for extracting the final answers from model outputs is described in algorithm 1.

System Role: <You are an expert tutor who solves multiple-choice questions by reasoning step by step.>

Task: <Explain the solution process step-by-step...>

Final Answer: [Letter]

Question: <question>

Options: <chr(10).join(options)>

Figure 2: LLM Prompt for Step-by-Step MCQ Explanation

Algorithm 1: Extracting Final Answers Using Prompt and Regex

```
Input: A set of multiple-choice questions
       with their choices
Output: Final answers extracted from
        model responses
foreach question in questions do
    Get the question text and its
      corresponding choices
    Create the full prompt by embedding
      the question and choices into the
      template
    Send the prompt to the language model
    Extract the final answer using regex
      (e.g., match Final Answer: [A-F])
```

In addition to the two main prompting strategies, we experimented with integrating Retrieval-Augmented Generation (RAG) using the available in-dataset context. Specifically, each chunk in the retrieval index consisted of a question–answer pair from the training data. At inference time, for each test question, the most similar chunk was retrieved and appended to the LLM prompt as supporting context.

We also estimated the inference cost of our approach. Running the full QIAS benchmark with step-by-step reasoning required approximately **10 USD** in API costs.

In addition to the CoT prompting strategy, we experimented with a retrieval-augmented generation (RAG) setup. We built the retrieval index directly from the training split, where each chunk was a **concatenation of the original question and an expanded, detailed answer** (i.e., *question + enriched answer*) to provide richer signals. Chunks were stored in a **ChromaDB** vector index. We evaluated several embedding models, including **Omartificial-Intelligence-Space/Arabic-Triplet-Matryoshka-V2**, **sentence-**

transformers/all-MiniLM-L6-v2, and **Begm3**; among these, **Begm3** yielded the strongest retrieval quality in our setting. At inference time, for each test question we **retrieved the top-3 nearest chunks** and appended them to the step-by-step reasoning prompt, then used **Claude 3.7** in CoT mode to select the final answer.

5 Results

The performance of the language models was evaluated with and without the application of a "thinking" step. Without thinking, Claude 3.7 achieved a score of 0.67, while GPT-4o scored slightly lower at 0.6312. When the thinking step was incorporated, performance significantly improved for both models. Claude 3.7 (with thinking) reached a score of 0.81, GPT-4o (with thinking) achieved a score of 0.74, which is similar to the RAG + Claude score. These results suggest that structured reasoning or "thinking" substantially enhances the capabilities of both models.

Model	Score
Claude 3.7	0.67
GPT-4o	0.63
Claude 3.7 (Thinking)	**0.81**
GPT-4o (Thinking)	0.74
RAG (context + Claude 3.7 Thinking)	0.737

Table 1: Performance scores of models with and without the thinking step, on the dev dataset

Model	Score
Claude 3.7	**0.71**
GPT-4o	0.55

Table 2: Performance scores of models with the thinking step, on the test dataset

The RAG+CoT configuration produced **accuracy close to, but lower than**, our CoT-only baseline on the dev set (**0.737** with RAG+CoT vs. **0.81** with CoT-only using Claude 3.7). While competitive, the RAG pipeline introduced additional complexity (sensitivity to embedding choice) and increased hallucination risk from injecting three retrieved passages (occasional reliance on partially relevant context). Given these trade-offs, and since CoT-only already outperformed it, we did **not** adopt RAG in the final system.

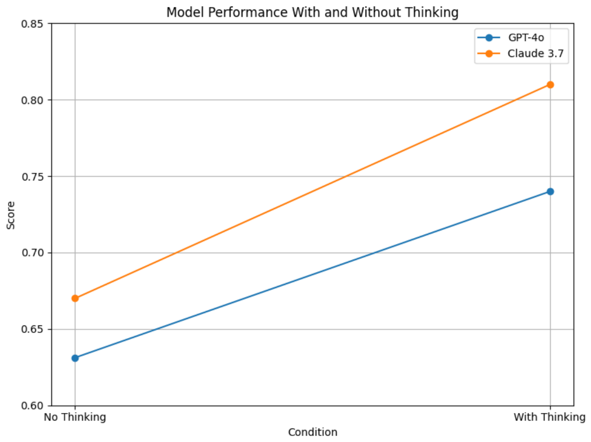


Figure 3: Comparison of model performance with and without thinking.

5.1 Error Analysis

Understanding the reasoning process and underlying thought patterns can reveal valuable insights and promising directions in the study of Chain of Thought reasoning (Wei et al., 2023). The following examples highlight specific instances where errors occur in the reasoning steps:

- First example: A person passed away and was survived by the following heirs: three full brothers, three sons of full brothers (nephews), four daughters, one wife, and five full sisters. The question is: what is the wife's share from the corrected total of the estate (al-tarkeha al-musahhah)?

The model (Claude) answer with thoughts was:

- Let us solve this inheritance problem step by step. First, we identify the heirs and their respective shares: the wife receives one-eighth (1/8) of the estate due to the presence of children; the four daughters collectively receive two-thirds (2/3) because they are multiple; the remaining estate goes to the full brothers and full sisters by way of residuary inheritance (taṣīb); while the sons of the full brothers are excluded due to the presence of closer male heirs. Next, we determine the original estate denominator, which is 24, since it accommodates both one-eighth and two-thirds. The distribution then becomes: 3 shares to the wife (1/8 of 24), 16 shares to the daughters (2/3 of 24), and the remaining 5 shares to be divided among the full brothers and sisters. However, since these 5 shares are to be divided among eight people (3 brothers and 5 sisters), the division is not exact, so we must correct the estate by finding a common multiple. We multiply the original 24 by 8 (the number of residuary heirs), resulting in a corrected estate of 192. The adjusted distribution is as follows: the wife receives 24 shares (3 × 8), the daughters receive 128 shares (16 × 8), and the full siblings receive 40 shares (5 × 8). Finally, since the question asks about the wife's share from the original estate before correction, her share is 3 out of 24. Thus, the correct answer is: A.

Analysis: The model answer was completly correct until reaching the 'Tasheeh' part where it did not write the shares after 'Tasheeh' so it chooses the before 'Tasheeh' answer.

- Second Example: A man died leaving: 3 sons, a mother, and 3 daughters. How many shares does each son receive?

The model (Claude) answer with thoughts was:

- Okay, let's solve this problem step by step: (1) Identify the heirs and their shares: the mother — one-sixth (because there are descendant heirs); the sons — the residue by ta'sīb (residuary), males receive twice the share of females; the daughters — together with the sons they inherit by ta'sīb with the male receiving twice the female's share. (2) Base of the calculation = 6 (the denominator for the sixth). (3) Distribution of shares: mother: 1 share (the sixth); the remainder 5 shares go to the sons and daughters by ta'sīb. (4) Calculate the number of portions: each son = 2 portions, each daughter = 1 portion, total = (3×2) + (3×1) = 9 portions. (5) Value of one portion: 5 (remaining shares) × 6 = 30 shares; 30 ÷ 9 = 10 shares per portion. (6) Each son's share: 2 × 10 = 20 shares. Looking at the available choices, we find each son's share is 20 shares. Final answer: F.

Analysis: Here the model did multiple errors from the first step of explaining the shares on this simpler question.

6 Future Work

Our RAG pilot with in-dataset QA context showed only limited improvements over CoT-only prompting, mainly due to coverage and retrieval granularity issues. To strengthen system robustness, future work will focus on three directions: (i) building a domain-specific inheritance corpus from authoritative sources to reduce retrieval sparsity, (ii) exploring structure-aware chunking and hybrid retrieval strategies to better capture fiqh-specific rules, and (iii) integrating RAG with CoT under context constraints to improve reasoning quality without incurring the overhead of fine-tuning.

7 Conclusion

In this paper, we presented our Chain-of-Thought prompting system for Islamic inheritance reasoning in the QIAS 2025 shared task. We introduced a dual-prompting approach that effectively addresses the complexity of Islamic jurisprudence reasoning through step-by-step explanation before answer selection. Our experiments demonstrate that incorporating a "thinking" step significantly improves model performance, with Claude 3.7 and GPT-4o achieving 21% and 17% relative improvements respectively. Through error analysis, we identified inheritance correction procedures (Tasheeh) as a primary area for future improvement, where models correctly perform initial calculations but fail to apply final correction steps. Our work confirms the difficulty of Islamic inheritance reasoning tasks but also shows that structured prompting can substantially enhance SOTA language models' performance on complex Arabic legal reasoning without additional model modifications or training.

References

Hazem Abdelazim, Tony Begemy, Ahmed Galal, Hala Sedki, and Ali Mohamed. 2024. Multi-hop arabic llm reasoning in complex qa. *Procedia Computer Science*, 244:66–75. 6th International Conference on AI in Computational Linguistics.

Manswab Mahsen Abdulrahman and Abdul Hafiz Musa Walusimbi. 2024. Evaluating the use of artificial intelligence for issuing fatwas in islamic inheritance cases: A juristic study with a comparison to gpt-3.5. *Asy-Syari'ah*, 26(2):121–146.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydın. 2025. Improving llm reliability with rag in religious question-answering: Mufassirqas. *Turkish Journal of Engineering*, 9(3):544–559.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 2025a. Qias 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Second Arabic*

Natural Language Processing Conference, ArabicNLP 2025, Suzhou, China, November 5–9, 2025. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 Second Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Serry Sibaee, Omer Nacar, Adel Ammar, Yasser Al-Habashi, Abdulrahman Al-Batati, and Wadii Boulila. 2025. From guidelines to practice: A new paradigm for arabic language model evaluation. *Preprint*, arXiv:2506.01920.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University - Computer and Information Sciences*, 33(1):68–76.

N&N at QIAS 2025: Chain-of-Thought Ensembles with Retrieval-Augmented framework for Classical Arabic Islamic MCQs

Nourah Alangari
King Saud University
nmalangari@ksu.edu.sa

Nouf AlShenaifi
King Saud University
noalshenaifi@ksu.edu.sa

Abstract

We present our system developed for the Question-and-Answer in Islamic Studies Assessment Shared Task on Evaluating LLMs for Islamic Knowledge (QIAS 2025), which focuses on answering Arabic multiple-choice questions (MCQs) derived from classical Islamic texts. Our methodology integrates few-shot chain-of-thought prompting across multiple LLMs, enhanced by a majority-vote ensemble mechanism. In situations of ensemble uncertainty, we deploy a retrieval-augmented re-prompting module that extracts contextually relevant passages from digitized Islamic sources to refine model predictions. Our final system achieves an accuracy of **89.8%** on the hidden test set.

1 Introduction

Recent advancements in large language models (LLMs) have significantly enhanced their capabilities in understanding and reasoning across diverse knowledge domains. However, their performance on specialized, culturally-rich content such as classical Islamic texts remains less explored. Classical Islamic texts—covering jurisprudence, creed, exegesis, and hadith—pose distinctive challenges: they are primarily in Arabic, employ specialized terminology, encode subtle doctrinal distinctions across legal schools, and often require multi-step reasoning (e.g., analogical and numerical reasoning in inheritance) to reach a correct answer (Bouchekif et al., 2025b). In this context, we present our system for the Question-and-Answer in Islamic Studies Assessment Shared Task on Evaluating LLMs for Islamic Knowledge (QIAS 2025) (Bouchekif et al., 2025a), which involves answering Arabic multiple-choice questions (MCQs) drawn specifically from classical Islamic literature. Our proposed approach integrates few-shot chain-of-thought prompting across several prominent LLMs, coupled with a robust majority-vote ensemble strategy. When the ensemble fails to reach consensus, our retrieval-augmented re-prompting (R^{2}P) module dynamically retrieves relevant textual evidence from digitized Islamic resources, enabling models to produce refined and contextually grounded predictions. Our final submission achieves an accuracy of **89.8%** on the hidden test set. The remainder of this paper is structured as follows: Section 2 reviews related work. Section 3 describes the QIAS 2025 task and dataset. Section 4 presents the system overview. Section 5 details the experimental setup. Section 6 reports and analyzes results. Section 7 concludes and outlines future work. An Appendix includes prompt templates and additional examples.

2 Related work

Large Language Models (LLMs) have shown remarkable advancements in zero-shot and few-shot reasoning tasks (Al Nazi et al., 2025) (Meshkin et al., 2024). Chain-of-thought (CoT) prompting has emerged as a powerful strategy to guide LLMs through intermediate reasoning steps before producing a final answer. Introduced by Wei et al. (Wei et al., 2022), CoT prompting significantly improved performance on tasks requiring logical reasoning, arithmetic, and commonsense inference. Later, Wang et al. (Wang et al., 2022) enhanced this framework with self-consistency sampling, where multiple reasoning paths are sampled, and the most consistent final answer is selected resulting in more robust predictions. While these techniques have been extensively evaluated on general-domain tasks in English, their application to Arabic particularly domain-specific Arabic such as classical Islamic jurisprudence and theology remains limited. Retrieval-augmented generation (RAG), introduced by Lewis et al. (Lewis et al., 2020), combines external document retrieval with generation-based models to inject relevant background knowl-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 878–882
November 8-9, 2025 ©2025 Association for Computational Linguistics

edge into the reasoning process. RAG has shown utility in open-domain QA, but few studies have adapted this method to classical Arabic corpora with domain-specific embeddings and passage reranking (Omoush and Ghnemat, 2025) (Bazzi and Gaith, 2025). Our contribution is novel in its application of CoT ensembles with on-demand retrieval for domain-specific Arabic MCQs—a setting that requires precise integration of theological and jurisprudential sources. This combination of retrieval-augmented CoT and ensemble majority voting is particularly impactful for advanced questions requiring deeper contextual grounding.

Example:
ما هو القول القديم للشافعي في صوم أيام التشريق؟

A) **لا يجوز صومها مطلقاً.**
B) **يجوز صومها للمتمتع إذا عدم الهدي عن الأيام الثلاثة الواجبة في الحج.**
C) **يجوز صومها لمن لم يجد الهدي فقط.**
D). **يجوز صومها للمسافر فقط**

Figure 1: A sample MCQ from QIAS 2025 Subtask 2.

3 Task Description

3.1 Task setup

The QIAS 2025 Subtask 2 involves answering classical Islamic multiple-choice questions (MCQs) in Arabic. Each input consists of a question stem and four possible answers (labeled A–D), with a single correct option. For example, a typical input might present a jurisprudential question derived from classical Islamic texts and require the system to output the correct choice label. This task requires deep semantic understanding, domain-specific expertise—particularly within Islamic contexts—and a keen ability to discern subtle linguistic nuances in the Arabic language.

3.2 Dataset

The dataset employed in this task comprises 1,400 Arabic multiple-choice questions (MCQs), evenly divided into 700 for validation and 700 for testing. These questions are meticulously curated from authoritative classical Islamic texts and cover a range of domains, including Fiqh (Islamic jurisprudence), Sīrah (the prophetic biography), Ulūm al-Qur'ān (Qur'anic sciences), and Ulūm al-Hadith (Hadith studies). To assess the system's reasoning capabilities, the questions are categorized into three levels of difficulty—Beginner, Intermediate, and Advanced—each reflecting a progressively deeper level of conceptual and analytical complexity (Bouchekif et al., 2025a). Additionally, well-known Islamic e-books such as Ar-Raḥīq al-Makhtūm (The Sealed Nectar) and Al-Itqān fī Ulūm al-Qur'ān (The Perfect Guide to the Sciences of the Qur'an) are provided as supplementary resources, serving as foundational references for the task. Figure 1 presents a sample multiple-choice question (MCQ) from the QIAS 2025 Shared Task (Subtask 2: Islamic Assessment), which evaluates language models' understanding of classical Islamic knowledge.

3.3 Track Participation

We participated in the QIAS 2025 Shared Task (Subtask 2: Islamic Assessment), part of the ArabicNLP 2025 conference held in conjunction with EMNLP 2025. This subtask centers on evaluating large language models (LLMs) in the domain of classical Islamic knowledge through multiple-choice questions. As one of the first benchmarks specifically designed for Arabic MCQs in religious and jurisprudential contexts, it provides a structured and rigorous framework for assessing deep semantic understanding and domain-specific reasoning in Islamic studies.

4 System Overview

Our pipeline, illustrated in (Figure 2), consists of three main stages designed for robust Arabic Islamic multiple-choice question answering:

1. **Prompt Sampling and Few-Shot CoT Prompting:** In the first stage, we leverage few-shot chain-of-thought (CoT) prompting techniques. Five carefully selected demonstration examples from the QIAS validation MCQs are embedded into a standardized Arabic prompt template.

2. **Majority Ensemble:** In the second stage, we employ a majority voting ensemble using the top three performing models selected from GPT-4o, Qwen-Plus, Gemini 2.5, and DeepSeek. For each instance, we collect the predictions from these three models and determine the final output based on majority agreement—specifically, a label is selected only

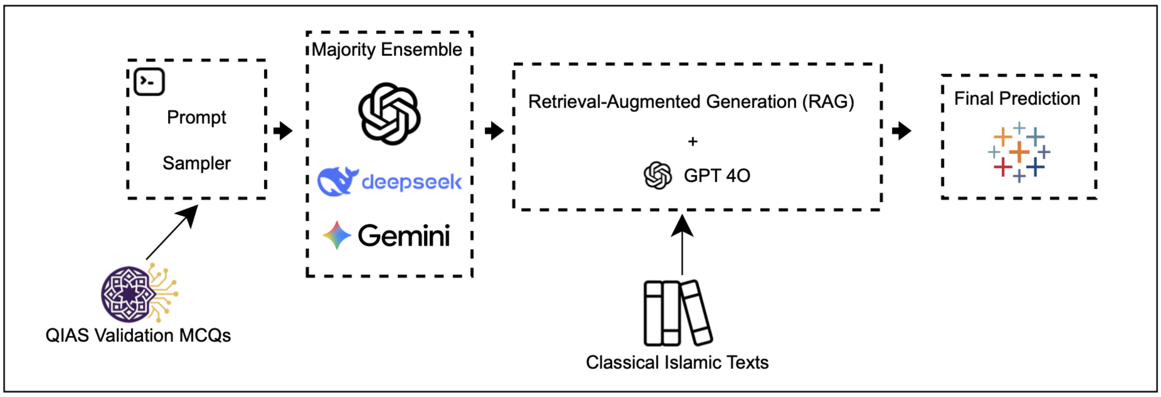


Figure 2: Overview of our ensemble-RAG pipeline combining LLMs and classical Islamic texts for answering QIAS 2025 MCQs.

if it is endorsed by at least two of the three models.

3. **Retrieval-augmented re-prompting (R^{2}P):** For cases where the ensemble stage results in uncertainty (i.e., no option reaches the required majority), we apply a retrieval-augmented re-prompting strategy. This approach involves:

    - Dense-only retrieval over classical Islamic texts: Arabic-LaBSE (768-d, mean-pooled, L2-normalized; inner-product) + FAISS IndexFlatIP on chunks 180–220 tokens (overlap 40–50) **retrieving the top-10** relevant passages.
    - Re-ranking these retrieved passages using a hybrid BM25 and cross-encoder scorer to select the top 3 most relevant passages.
    - Re-prompting the GPT-4o model with these carefully selected passages to produce a refined final prediction.

5 Experimental Setup

Data splits. We used the official dataset provided by the organizers, comprising 700 validation items and 700 test items, without additional splitting.

Hyper-parameters. To promote diversity while maintaining coherence in generation, we adopt the following settings: temperature = 0.2, top-p = 0.95, and a maximum of 512 output tokens.

Models considered. We evaluate the following models: GPT-4o, Gemini 2.5-Flash, Qwen-Plus, and DeepSeek-V3 [1]. Only the top three performers are included in the ensemble.

Evaluation metrics. We evaluate performance solely based on accuracy, measured as the percentage of questions for which the model's prediction exactly matches the correct answer, using the official **Task2_MCQ_Test_gold_labels** provided by the organizers.

6 Results

The performance of the evaluated models under different learning scenarios (zero-shot, 3-shot, and 5-shot) is summarized in Table 1. GPT-4o consistently demonstrated strong results across all settings, with a slight improvement observed in the 5-shot scenario. Gemini 2.5 exhibited a substantial performance increase when moving from zero-shot to few-shot learning conditions. This notable improvement can be attributed mainly to Gemini's initial difficulty in strictly adhering to task instructions in the zero-shot setting. Despite clear directives—such as prompts explicitly stating, "`Final Answer (letter only [A, B, C, D]) DO NOT output your thinking process or any other text except [A, B, C, D]:`"—Gemini often generated excessively detailed outputs, frequently exceeding the maximum token limit, leading to incomplete or empty responses. However, providing few-shot examples substantially improved Gemini's ability to comply with the task requirements, resulting in competitive accuracy. DeepSeek and Qwen-Plus also showed consistent improvement with the increase in examples provided, though

[1] All models were accessed via their official APIs between 15 - 20 July 2025.

their overall performance lagged slightly behind GPT-4o and Gemini, particularly in the few-shot scenarios. Our proposed system achieved an accuracy of 0.90 in the 5-shot setting, surpassing all individual models tested. This highlights the effectiveness of integrating few-shot prompting, model ensembling, and retrieval-augmented re-prompting. By combining the complementary strengths of multiple models and addressing uncertainty through targeted retrieval and refined prompting, our system demonstrates greater accuracy and robustness than any single model operating independently.

Model	Zero-shot	3-shot	5-shot
GPT-4o	0.85	0.85	0.86
Gemini 2.5	0.59	0.87	0.87
DeepSeek-V3	0.79	0.80	0.84
Qwen-Plus	0.77	0.77	0.78
Our Model			0.898

Table 1: Performance comparison of four models under zero-shot, 3-shot, and 5-shot settings. Scores are approximated to two decimal places.

7 Conclusion

In this study, we effectively combined few-shot chain-of-thought prompting, a majority-vote ensemble strategy, and retrieval-augmented re-prompting to address the challenging task of answering classical Arabic Islamic multiple-choice questions (MCQs). Our proposed system achieved superior performance, demonstrating the effectiveness of integrating ensemble strategies with retrieval methods for domain-specific knowledge tasks.

Acknowledgments

We would like to thank the organizers of the QIAS 2025 Shared Task for providing valuable resources and guidance throughout the competition. Special thanks to anonymous reviewers for their constructive comments, which helped improve the quality of this work.

References

Zabir Al Nazi, Md Rajib Hossain, and Faisal Al Mamun. 2025. Evaluation of open and closed-source llms for low-resource language with zero-shot, few-shot, and chain-of-thought prompting. *Natural Language Processing Journal*, 10:100124.

Wafa Bazzi and Mervat Gaith. 2025. The wonders of rag: Streamlining knowledge with advanced techniques systematic literature review report. Technical report.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 2025a. Qias 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Second Arabic Natural Language Processing Conference, Arabic-NLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 Second Arabic Natural Language Processing Conference (Arabic-NLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Hamed Meshkin, Joel Zirkle, Ghazal Arabidarrehdor, Anik Chaturbedi, Shilpa Chakravartula, John Mann, Bradlee Thrasher, and Zhihua Li. 2024. Harnessing large language models' zero-shot and few-shot learning capabilities for regulatory research. *Briefings in Bioinformatics*, 25(5):bbae354.

Ebtehal H. Omoush and Rawan Ghnemat. 2025. Advancing arabic medical question answering systems with rag and llms integration. In *Proceedings of the International Conference on New Trends in Computing Sciences (ICTCS)*, pages 511–516. IEEE.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed 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. *In Proceedings of the Advances in Neural Information Processing Systems*, 35:24824–24837.

A Appendix

Prompt template:

```
“You are an Islamic knowledge expert
```

tasked with solving multiple-choice questions. Think step-by-step, carefully justify your reasoning, and then select the correct answer clearly.Here are some examples to follow:

مثال ١: السؤال: ما هو القول القديم للشافعي في صوم أيام التشريق؟
الخيارات:
A) لا يجوز صومها مطلقاً.
B)يجوز صومها للمتمتع إذا عدم الهدي عن الأيام
C)الثلاثة الواجبة في الحج. يجوز صومها لمن
D)لم يجد الهدي فقط. يجوز صومها للمسافر
فقط.

خطوات التفكير:

أيام التشريق هي الأيام الثلاثة التي تلي يوم النحر (يوم عيد الأضحى) في موسم الحج.

القولِ القدِّيم للشافعي يشير إلى رأيه الأول قبل أن يُرجّح عليه قوله الجديد، وقد يختلف عنه في بعض المسائل.

القول الجديد والمعتمد عند الشافعية هو تحريم صوم أيام التشريق مطلقاً، حتى للمتمتع إذا لم يجد الهدي، وهذا هو القول الذي عليه المتأخرون من أصحاب الشافعي.

في كتب الفقه على المذهب الشافعي، يُذكر أن من أقواله القديمة جواز صيام أيام التشريق في حالات مخصوصة.

من تلك الحالات: أن يكون الحاج متمتعاًا (أي جمع بين العمرة والحج في سفرة واحدة مع فاصل)، ولم يِجد الهدي، فيلزمه صيام ثلاثة أيام في الحج، ويُسمح له بصيامها في أيام التشريق.

هذا القول موثق ومعروف في كتب المذهب عند الحديث عن اختلاف القولين القديم والجديد للشافعي.

الإجابة النهائية: B

⋮

... the other 4 examples...

⋮

Now answer the question below by selecting the appropriate answer:
Question: {question}
Choices: {formatted_options}
Final Answer (letter only [A, B, C, D]) DO NOT output your thinking process:"

HIAST at QIAS 2025: Retrieval-Augmented LLMs with Top-Hit Web Evidence for Arabic Islamic Reasoning QA

Mohamed Motasim Hamed[*], **Riad Sonbol**[*], **Nada Ghneim**[**]
[*]Higher Institute for Applied Sciences and Technology, Damascus, Syria
[**]Arab International University, Daraa, Syria
{motasim.hamed, riad.sonbol}@hiast.edu.sy, n-ghneim@aiu.edu.sy

Abstract

We describe our participation in the QIAS 2025 Shared Task on Islamic Studies Question Answering, comprising two subtasks: (1) Islamic Inheritance Reasoning and (2) General Islamic Knowledge Assessment. Both were solved using the Claude 4 Opus LLM via API with tailored prompting. For Subtask 1, we implemented a lightweight Retrieval-Augmented Generation (RAG) pipeline, which retrieves the top Google Search result (often from IslamWeb), preprocesses it, and appends it to a structured few-shot Arabic prompt, thereby boosting reasoning accuracy. For Subtask 2, where web-retrieval was not feasible due to closed-book sources, we applied topic-diverse few-shot prompting to leverage the model's internal knowledge. Our systems achieved 4th/15 (0.895) in Subtask 1 and 3rd/10 (0.9259) in Subtask 2, demonstrating the effectiveness of targeted retrieval in open-web contexts and structured prompting in closed-domain Arabic QA.

1 Introduction

The QIAS 2025 Shared Task (*Question and Answer in Islamic Studies Assessment*) serves as a benchmark for evaluating large language models (LLMs) on domain-specific reasoning in Islamic knowledge (Bouchekif et al., 2025a). It consists of two multiple-choice subtasks: (1) Islamic Inheritance Reasoning (ᶜIlm al-Mawārīth) and (2) Islamic Knowledge Assessment (covering Fiqh, al-Ḥadīth, Tafsīr, uṣūl al-fiqh, etc.), with MCQs spanning beginner, intermediate, and advanced levels—designed to assess reasoning accuracy in both retrieval-supported and retrieval-free settings.

The task is conducted entirely in Arabic, reflecting the primary language of Islamic scholarship and presenting a significant challenge for LLMs given the language's morphological richness and syntactic complexity.

The two subtasks differ in their source and the feasibility of web-based retrieval. Subtask 1 draws primarily from online fatāwā, making retrieval from the open web practical and often effective. Subtask 2, by contrast, is based on classical and modern Islamic closed books that are generally not available through open web indexing; although retrieval may be beneficial in this task, we opted to approach it using the internal knowledge of the language model.

Our submission addresses both QIAS subtasks using LLM-based pipelines built around the Claude Opus 4 API. For Subtask 1 (Retrieval-supported QA), we adopted a single-document retrieval approach using the Google Search API, appending the top-ranked result, often from IslamWeb, to a structured few-shot prompt. This provided the model with both contextual exemplars and relevant retrieved knowledge, substantially enhancing rule-based reasoning in inheritance scenarios. Across various strategies and LLMs, supplementing questions with retrieved content from reliable sources yielded accuracy improvements exceeding 14% over using the model's internal knowledge alone.

For Subtask 2 (Zero-retrieval QA), where the data source comprised offline books, no external retrieval was performed. Instead, we employed a structured few-shot prompt with curated exemplars, leveraging the model's internal knowledge and reasoning capabilities. This strategy achieved competitive accuracy, demonstrating the model's ability to recall domain-specific information and perform sophisticated reasoning even in the closed-book setting.

In general, our findings underscore the effectiveness of structured web-retrieval in low-resource domains, such as Islamic law. QIAS offers a valuable benchmark for testing such approaches. We report the leaderboard results [1], along with the implemen-

[1] https://sites.google.com/view/qias2025/leaderboards

Proceedings of The Third Arabic Natural Language Processing Conference, pages 883–891
November 8-9, 2025 ©2025 Association for Computational Linguistics

tation details [2].

2 Background

2.1 Task Setup

We participated in **both** subtasks of the QIAS 2025 Shared Task on evaluating multilingual LLMs in Islamic reasoning and knowledge, as described in the task overview (Bouchekif et al., 2025a).

Subtask 1: Islamic Inheritance Reasoning (ʿIlm al-Mawārīth) Each item in Subtask 1 frames a detailed family fact pattern involving heirs such as wife, parents, full or half siblings, children, or deceased heirs. A multiple-choice question (6 options; exactly one correct) asks for the appropriate heir-share(s) based on fixed-rule Islamic jurisprudence (farāʾiḍ).

Example of Beginner level in Arabic, from the official dataset:

توفي عن أب، و2 أخ شقيق، و1 ابن أخ شقيق، و2 عم شقيق للأب، وأم، و2 بنت، و1 زوجة، ما هو نصيب الأم؟
A) الثلث, B) الربع, C) السدس, D) الثمن, E) النصف, F) لا شيء

Subtask 2: Islamic Knowledge Assessment Subtask 2 contains knowledge-based exam MCQs on topics such as ʿulūm al-Qurʾān, al-Ḥadīth, fiqh, uṣūl al-fiqh, sīrah, and Aqīdah, designed to elicit doctrinal, doctrinal-reasoned, or interpretive recall. The questions come in 4-option MCQ format, with exactly one correct answer.

Example of Beginner level in Arabic, from the official dataset:

ما مدة المسح على الخفين للمقيم؟
A)يوم وليلة, B)ثلاثة أيام بلياليهن, C) يومان وليلتان, D) أسبوع كامل

2.2 Dataset Details

Subtask 1: The dataset comprises ~20,000 training, 1,000 validation, and 1,000 test MCQs (six options each), generated from curated IslamWeb fatwas via Gemini 2.5 and validated by domain experts. Pre-processing involved deduplication and disambiguation. An auxiliary corpus of 3,165 original fatwas was also provided as an optional knowledge base (Bouchekif et al., 2025a).

Subtask 2: A collection of classical Islamic texts was provided as unsupervised data to fine-tune or as part of a Retrieval-Augmented Generation (RAG) (Lewis et al., 2020). From 25 reference texts, 1,400 MCQs (700 validation, 700 test) were created in seven disciplines (beginner-advanced), each with four answer choices, and validated by five experts.

2.3 Prior Work

Recent advances in Arabic LLMs, such as Jais (Sengupta et al., 2023), AceGPT (Huang et al., 2023), ALLaM (Bari et al., 2024), Kuwain (Hennara et al., 2025), and Fanar (Abbas et al., 2025) have expanded the ability to understand religious texts by pretraining large-scale corpora, including Quran, Hadith, and fatwa archives. Shared tasks have emerged to benchmark Islamic NLP, including Quranic QA (Malhas et al., 2022, 2023), Islamic knowledge retrieval (Qamar et al., 2024), and most recently QIAS 2025 (Bouchekif et al., 2025a), which evaluates LLMs on Islamic inheritance reasoning and general religious knowledge. Prior work in automating Islamic inheritance (IRTH) largely relied on expert systems (Akkila and Naser, 2016; Tabassum et al., 2019; Zouaoui and Rezeg, 2021) that encoded symbolic rules, or on RAG-based QA pipelines for Islamic texts (Alan et al., 2024; Sayeed et al., 2025). However, the performance of LLM on Islamic content remains constrained by factual errors, misinterpretation of context, and sensitivity to question phrasing (Mohammed et al., 2025; Alnefaie et al., 2023; Bouchekif et al., 2025b).

In Subtask 1, we used a lightweight, single-document retrieval-augmented generation setup, grounding Claude Opus 4 with authoritative sources retrieved via the Google Search API. This lightweight approach avoids handcrafted rules and instead provides juristic context for in-context reasoning. By contrasting this retrieval-supported configuration with a zero-retrieval baseline in Subtask 2, we enable a controlled comparison of retrieval-augmented versus unsupported Islamic-domain reasoning.

3 System Overview

Our QIAS 2025 submission adopts two distinct configurations, each tailored to its subtask, both powered by the Claude Opus 4 API.

3.1 Subtask 1: Islamic Inheritance Reasoning (Lightweight RAG)

We implemented a single-document Retrieval-Augmented Generation (RAG) pipeline. Each question was paired with the top-ranked Google Search

[2] https://gitlab.com/Moatasem444/qias2025-hiast-submission/

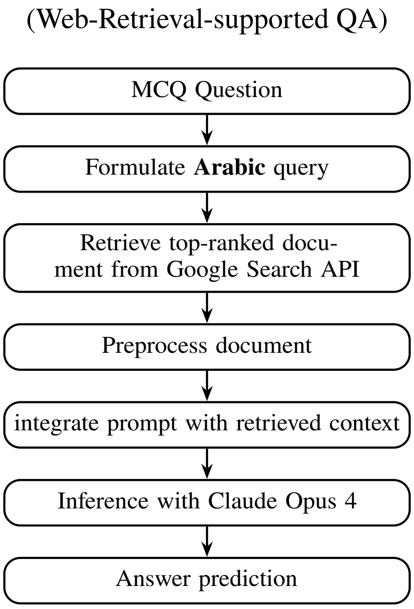

(Web-Retrieval-supported QA)
MCQ Question
Formulate **Arabic** query
Retrieve top-ranked document from Google Search API
Preprocess document
integrate prompt with retrieved context
Inference with Claude Opus 4
Answer prediction

Figure 1: Web-Retrieval pipeline for Subtask 1.

result (primarily from IslamWeb[3], pre-processed to remove boilerplate and appended (in Arabic) to the questions and answer choices. This method provided high-quality domain-specific grounding while avoiding the latency and complexity of multi-document vector retrieval. The complete retrieval pipeline is provided in figure 1.

3.2 Subtask 2: Islamic Knowledge Assessment (Zero-Retrieval Few-Shot)

Since web-retrieval was not possible, we adopted a few-shot prompting strategy. Three to four representative MCQs, covering different topics and difficulty levels, were inserted into a fixed Arabic prompt template before the test question. This few-shot configuration leveraged the LLM's internal knowledge for doctrinal and interpretive reasoning. Detailed work examples are provided in figure 2. The detailed prompt formulations corresponding to the two subtasks are provided in Appendix A.

3.3 Challenges Addressed

We address several challenges in our approach, including Arabic morphology and orthography, where queries preserve diacritics to improve retrieval precision; input length control, with retrieved passages truncated to $L_{\max} = 2000$ characters to fit model limits; and knowledge coverage, where few-shot examples are selected for topical diversity to reduce bias toward frequent topics.

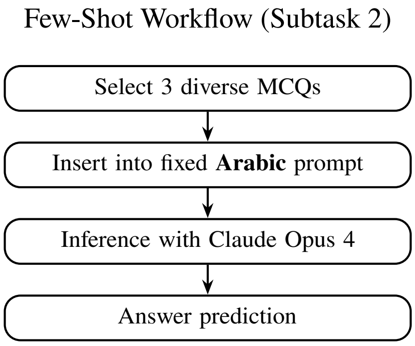

Few-Shot Workflow (Subtask 2)
Select 3 diverse MCQs
Insert into fixed **Arabic** prompt
Inference with Claude Opus 4
Answer prediction

Figure 2: Few-shot pipeline for Subtask 2.

4 Experimental Setup

4.1 Data Sources and Preprocessing

For **Subtask 1**, we performed real-time retrieval from publicly available web sources, primarily IslamWeb, using the Google Custom Search API, instead of using the 3,165 IslamWeb fatwas provided. The retrieved documents were cleaned by removing HTML tags and boilerplate text and truncated to $L_{\max} = 2000$ UTF-8 characters.

For **Subtask 2**, the large corpus of classical Islamic texts was not used. Instead, we applied a few-shot prompt with $k = 3$ examples drawn from the validation set (to prevent data leakage). The few-shot MCQs were normalized to a consistent template format comprising an Arabic question stem and four labeled options.

4.2 Task Evaluation Metrics

We followed the official QIAS 2025 evaluation protocol (Bouchekif et al., 2025a), using only the validation and test sets provided. The precision in the test set, calculated as the proportion of exact matches between the predictions and the gold answers, was the only ranking metric. The outputs were normalized to choice letters. The source code is available online[4].

4.3 Implementation Details

Hyperparameters. Both configurations used the default Claude Opus 4 API parameters, with temperature set to 0.0 for deterministic outputs and max_tokens fixed at 1000. For Subtask 2, the number of few-shot exemplars was set to $k = 3$ based on preliminary validation and the number of difficulty levels.

[3] https://www.islamweb.net

[4] https://gitlab.com/Moatasem444/qias2025-hiast-submission/

External Tools and Libraries. We employed the Claude Opus 4 API (Anthropic, May 2025 release) as the primary LLM, accessed via its paid subscription tier. Live web-retrieval for Subtask 1 was performed using the Google Custom Search API[5], also under a paid usage plan. All API integration and pre-processing were implemented in Google Colab.

5 Results

5.1 Official Leaderboard Performance

We achieved strong results in test data for both subtasks, placing 4th/15 in Subtask 1 (Accuracy: 0.895) and 3rd/10 in Subtask 2 (Accuracy: 0.9259). Detailed scores and rankings are shown in Table 5 (Appendix B).

5.2 Comparative Analysis and Error Patterns

For Subtask 1, we compared our lightweight Google Search API pipeline with:

(a) the same LLM without retrieval,

(b) the same LLM with its built-in "web search" mode.

Our approach consistently outperformed all baseline methods in a wide range of models, including closed-source systems such as Claude 4 Opus (C4O), GPT 4.1 Mini (G4.1M), and Gemini 2.5 Flash (G2.5F), as well as the open-source model like Fanar (Table 1). In particular, even minimal, yet high-quality retrieval yielded substantial gains in accuracy.

In contrast, for Subtask 2, where sources are closed-book, web-retrieval offered no benefit; structured few-shot prompting proved most effective (Table 2).

Table 1: Validation results for Subtask 1 under different models and retrieval settings.

Model	No Ret.	Built-in WS	Ours
C4O	0.785	0.812	**0.924**
G2.5F	0.700	N/A	0.871
G4.1M	0.580	0.690	0.822
Fanar	0.574	N/A	0.645

Error analysis on 50 random misclassifications per task revealed:

- **Task 1:** Failures occurred when the retrieval was missing or contained partial matches, forcing reliance on internal knowledge of LLM. Some models (e.g., Gemini) deviated from the format by including explanations. Other causes included ambiguous fatwā phrasing, missing numeric details, and context length limits ($L_{\max} = 2000$).
- **Task 2:** Most errors stemmed from fine-grained doctrinal differences and narrations that required exact recall. It should be noted that error rates are distributed inversely between difficulty levels.

Table 2: Validation results for Subtask 2 on Gemini 2.5 flash, showing no gain from web-retrieval (WebR).

Method	Acc.	Δ vs. Few-Shot
Few-Shot only	0.875	—
Few-Shot + WebR	0.805	−7%

5.3 Similarity and Prediction Accuracy

We computed the similarity score between text and the result of the top hits on the Web using the Muffakir Embedding model [6]. We then analyzed the relationship between the similarity score and the prediction precision. The results do not indicate significant relevance: the average similarity for incorrect predictions was 0.645, while for correct predictions it was 0.653, with a correlation of only 0.027. Importantly, it also indicates that retrieving only the top-ranked search result is sufficient: If the answer is present in the retrieved context, it is most likely in the first result, and additional results are unlikely to improve accuracy. This further supports the effectiveness of Google Search's ranking in providing the most relevant information for this task.

5.4 Error Analysis

We examined the relationship between the availability of the retrieved web context and the system error rates (Table 3). Although the number of cases with the retrieved web context (923) is substantially higher than those without (77), the relative error rate is lower (7.37% vs. 9.09%). This demonstrates the effectiveness of web-retrieval; If comparable contextual information had been available for the remaining instances, the overall error rate could have been reduced by up to 1.7%. In contrast, the absence of such context correlates with increased error rates. In particular, the system fails to retrieve relevant context in approximately 8.3% of the total cases, which directly limits the attainable performance limit. Reducing this retrieval failure rate is

[5] https://developers.google.com/custom-search

[6] https://huggingface.co/mohamed2811/Muffakir_Embedding

therefore critical to achieving consistently higher accuracy.

Table 3: Subtask 1 error rates with/without web context (NW = No Web).

Context	#Q	#Wrong	Err. (%)
With Web	923	68	7.37
NW	77	7	9.09

We also analyzed incorrect predictions by difficulty level of the questions in the validation sets for both tasks (Table 4). In task 1, the majority of errors occurred at the advanced level (45 errors, 60%), followed by the beginner level (30 errors, 40%). In Task 2, errors were more evenly distributed: the beginner level questions accounted for 40 errors (43%), intermediate for 31 errors (33.3%), and advanced for 22 errors (23.7%). These results suggest that in Task 1, advanced-level questions are disproportionately challenging, while in Task 2, errors are less skewed toward a single difficulty level, indicating a more balanced difficulty distribution.

Table 4: Wrong predictions by difficulty level in validation data. Percentages relative to total wrong predictions per task.

Level	Task 1		Task 2	
	Wrong	%	Wrong	%
Beginner	30	40.0	40	43.0
Intermediate	–	–	31	33.3
Advanced	45	60.0	22	23.7

Additional error samples are shown in Appendix C.

6 Conclusion

We addressed the QIAS 2025 Shared Task using large language models with task-specific prompting strategies. For Subtask 1, live Google Search retrieval achieved 0.895 accuracy, while for Subtask 2, few-shot prompting reached 0.9259 accuracy. The main limitations include the dependence on the quality of the retrieval, the doctrinal differences, and the dependence on the closed-source Claude Opus 4 API. Future work will fine-tune Arabic-specific models and employ domain-restricted RAG over curated texts to mitigate coverage gaps and ambiguity.

7 Acknowledgments

We acknowledge MISRAJ.AI for supporting parts of the development process and anonymous reviewers for their constructive comments.

References

Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur A Chowdhury, Fahim Dalvi, and 1 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *CoRR*.

Alaa N. Akkila and Samy S. Abu Naser. 2016. Proposed expert system for calculating inheritance in islam.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydin. 2024. A rag-based question answering system proposal for understanding islam: Mufassirqas llm. *arXiv preprint arXiv:2401.15378*.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Is gpt-4 a good islamic expert for answering quran questions? In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 124–133.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan AlRashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, and 1 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Abdessalam Bouchekif, Samer Rashwani, Emad Mohamed, Mutaz Al-Khatib, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Khalil Hennara, Sara Chrouf, Mohamed Motaism Hamed, Zeina Aldallal, Omar Hadid, and Safwan AlModhayan. 2025. Kuwain 1.5 b: An arabic slm via language injection. *arXiv preprint arXiv:2504.15120*.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Juncai He, and 1 others. 2023. Acegpt, localizing large language models in arabic. *arXiv preprint arXiv:2309.12053*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

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 *Proceedinsg 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.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Marryam Yahya Mohammed, Sama Ayman Ali, Salma Khaled Ali, Ayad Abdul Majeed, and Ensaf Hussein Mohamed. 2025. Aftina: enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.

Faiza Qamar, Seemab Latif, and Rabia Latif. 2024. A benchmark dataset with larger context for non-factoid question answering over islamic text. *arXiv preprint arXiv:2409.09844*.

Mohammad Amaan Sayeed, Mohammed Talha Alam, Raza Imam, Shahab Saquib Sohail, and Amir Hussain. 2025. From rag to agentic: Validating islamic-medicine responses with llm agents. *arXiv preprint arXiv:2506.15911*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Sadia Tabassum, AHM Hoque, Sharaban Twahura, and Mohammad Osiur Rahman. 2019. Developing an islamic farayez system applying software engineering. *Jurnal Kejuruteraan*, 31(1):25–38.

Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University-Computer and Information Sciences*, 33(1):68–76.

8 Appendix

A Prompt Templates

Subtask 1 Prompt

These are a few-shot examples for the task: answering multiple-choice questions by selecting the correct option.

Example 1:
Question: توفي عن أب، وأخوين شقيقين، وابن أخ شقيق، وعمين شقيقين، وأم، وبنتين، و زوجة، فما نصيب الأم؟
A) الثلث
B) الربع
C) السدس
D) الثمن
E) النصف
F) لا شيء
Answer: C

Example 2:
Question: توفي عن أخ من الأم، وبنت، وزوجة، وأختين من الأم: كم عدد أسهم البنت بعد الرد؟
A) سهم واحد
B) سهمان
C) ثلاثة أسهم
D) أربعة أسهم
E) سبعة أسهم
F) ثمانية أسهم
Answer: E

Context for the question: {context}

You are a specialist in Islamic sciences. Your task is to answer multiple-choice questions by selecting the correct option.

Question: {question} {options_text}

Please respond using **only one English letter** from the following: A, B, C, D, E, F.
Do not write any explanation or additional text.

Subtask 2 Prompt

These are a few-shot examples for the task: answering multiple-choice questions by selecting the correct option.

Example 1:
Question: ما مدة المسح على الخفين للمقيم؟
A) يوم وليلة
B) ثلاثة أيام بلياليهن
C) يومان وليلتان
D) أسبوع كامل
Answer: A

Example 2:

Question:

من شروط الأصل في القياس؟
A) أن يكون الأصل فرعا لأصل آخر
B) أن لا يكون الحكم ثابتا
في الأصل بطريق سمعي شرعي C) ألا يكون الأصل فرعا لأصل آخر
D) ألا تعرف طريقة الاستنباط

Answer: C

Example 3:

Question:

ما هو طريق الحكماء لإثبات وجود الواجب؟
A) عن طريق اعتبار العالم قديمًا.
B) عن طريق إثبات أن العالم واجب لذاته.
C) عن طريق امتناع التسلسل والدور.
D) عن طريق إثبات حدوث العالم.

Answer: C

You are a specialist in Islamic sciences. Your task is to answer multiple-choice questions by selecting the correct option.

Question: {question} {options_text}

Please respond using **only one English letter** from the following: A, B, C, D.
Do not write any explanation or additional text.

B Supplementary Results

Table 5: QIAS 2025 official leaderboards. Our system **HIAST** ranked **4th** in Subtask 1 (Acc. 0.895) and **3rd** in Subtask 2 (Acc. 0.9259) on the test set.

Subtask 1: Islamic Inheritance Reasoning (ranked by test Accuracy)

Rank	Team	Accuracy	Affiliation(s)
1	Gumball	0.972	Alexandria University, Ain Shams University
2	PuxAI	0.957	VNU-HCM University of Information Technology
3	NYUAD	0.927	New York University Abu Dhabi
4	**HIAST**	**0.895**	**Higher Institute for Applied Sciences and Technology**
5	MorAI	0.880	International Center for AI, Mohammed VI Polytechnic University

Subtask 2: Islamic Knowledge Assessment (ranked by test Accuracy)

Rank	Team	Accuracy	Affiliation(s)
1	PuxAI	0.9369	VNU-HCM University of Information Technology
2	Athar	0.9272	University of Khartoum, International Islamic University Malaysia
3	**HIAST**	**0.9259**	**Higher Institute for Applied Sciences and Technology**
4	N&N	0.8984	King Saud University
5	Tokenizers United	0.8738	Nile University

C Error Examples

These examples illustrate that Subtask 2 errors arise from doctrinal differences across Islamic disciplines (e.g., Sufism) as well as reference/attribution issues that require reliable sourcing.

Table 6: Examples of wrong predictions in Subtask 2 validation data, categorized by difficulty level and error type.

Level	Question (Arabic)	Correct Answer	Model Prediction	Error Type
Beginner	ما الذي يقصد بالفناء الصوفي؟	A) استغراق النفس في الروح الإلهي	B) موت النفس	Comprehension/Disambiguation
Intermediate	ما هو الحرف الناقص في آية سورة ص مقارنة بآية سورة البقرة؟	B) حرف "الواو"	A) حرف "أبى"	Comprehension/Disambiguation
Advanced	ما هو القول القديم للشافعي في صوم أيام التشريق؟	B) يجوز صومها للمتمتع إذا عدم الهدي عن الأيام الثلاثة الواجبة في الحج	A) لا يجوز صومها مطلقاً	Doctrinal Variance
Beginner	ما المصادر غير الإسلامية التي يرى ترمنجهام أن التصوف يمت إليها بصلة طفيفة؟	C) الحياة الصوفية الزهدية للمسيحية الشرقية	B) الفلسفة الهندية	Reference/Attribution Error
Advanced	ما هو الاعتراض الذي قد يورد على مقدمة إمكان مخالفة أحد الإلهين للآخر؟	C) أن مخالفة أحدهما للآخر وإرادة ضده ليست ممكنة دائمًا	B) أن المخالفة بين الإلهين مستحيلة	Doctrinal Variance
Beginner	هل تعرض الإمام البخاري في "التاريخ الكبير" للجرح والتعديل؟	B) تعرض أحياناً	C) تعرض دائماً	Reference/Attribution Error
Intermediate	ما معنى "اللمزة" في اللغة؟	B) الذي يعيب الناس سرًا ويؤذيهم	A) الذي يشتم الرجل علانية ويكسر عينيه عليه	Comprehension/Disambiguation

QU-NLP at QIAS 2025 Shared Task: A Two-Phase LLM Fine-Tuning and Retrieval-Augmented Generation Approach for Islamic Inheritance Reasoning

Mohammad AL-Smadi
Qatar University
Doha, Qatar
malsmadi@qu.edu.qa

Abstract

This paper presents our approach and results for SubTask 1: Islamic Inheritance Reasoning at QIAS 2025, a shared task focused on evaluating Large Language Models (LLMs) in understanding and reasoning within Islamic inheritance knowledge. We fine-tuned the Fanar-1-9B causal language model using Low-Rank Adaptation (LoRA) and integrated it into a Retrieval-Augmented Generation (RAG) pipeline. Our system addresses the complexities of Islamic inheritance law, including comprehending inheritance scenarios, identifying eligible heirs, applying fixed-share rules, and performing precise calculations. Our system achieved an accuracy of 0.858 in the final test, outperforming other competitive models such as, GPT 4.5, LLaMA, Fanar, Mistral and ALLaM evaluated with zero-shot prompting. Our results demonstrate that QU-NLP achieves near state-of-the-art accuracy (85.8%), excelling especially on advanced reasoning (97.6%) where it outperforms Gemini 2.5 and OpenAI's o3. This highlights that domain-specific fine-tuning combined with retrieval grounding enables mid-scale Arabic LLMs to surpass frontier models in Islamic inheritance reasoning.

1 Introduction

The rapid advancements in Large Language Models (LLMs) have opened new avenues for their application across diverse domains, including specialized knowledge systems. This paper details our participation in the QIAS 2025 Shared Task, specifically focusing on Subtask 1: Islamic Inheritance Reasoning (*Ilm al-Mawārīth*) (Bouchekif et al., 2025a). This subtask challenges LLMs to navigate the intricate and highly structured field of Islamic inheritance law, which is governed by precise jurisprudential rules. The objective is to develop systems capable of comprehending complex inheritance scenarios, accurately identifying eligible and ineligible heirs, applying fixed-share rules (*farāiḍ*), managing residuary shares, and addressing advanced cases such as proportional reduction (*ʿawl*) and redistribution (*radd*), ultimately performing precise calculations to determine final shares (Mohammedi, 2012; Zouaoui and Rezeg, 2021).

The intersection of Natural Language Processing (NLP) and legal reasoning, particularly within specialized domains like Islamic law, has garnered increasing attention. Prior research has explored the application of computational methods to analyze legal texts, extract relevant information, and even automate aspects of legal decision-making. However, the unique complexities of Islamic inheritance law, with its intricate rules and diverse scenarios, present distinct challenges for traditional NLP approaches (Malhas et al., 2022, 2023).

Recent advancements in Large Language Models (LLMs) have shown promising capabilities in complex reasoning tasks, including those requiring domain-specific knowledge. Studies have demonstrated LLMs' ability to understand and generate human-like text, perform question answering, and even engage in logical inference. However, their performance in highly specialized and rule-based domains often necessitates fine-tuning or integration with external knowledge sources (Almazrouei et al., 2023; Sengupta et al., 2023; Alnefaie et al., 2023; Bari et al., 2024; Mohammed et al., 2025).

Specifically, in the context of Islamic inheritance reasoning, several works have emerged (Akkila and Naser, 2016; Tabassum et al., 2019; Zouaoui and Rezeg, 2021). For instance, (Bouchekif et al., 2025b) assesses LLMs on Islamic legal reasoning, providing evidence from inheritance law evaluation. This work highlights the potential and limitations of current LLMs in this domain, underscoring the need for more robust and accurate systems.

Furthermore, the concept of Retrieval-Augmented Generation (RAG) has gained prominence as a method to enhance LLM

Proceedings of The Third Arabic Natural Language Processing Conference, pages 892–898
November 8-9, 2025 ©2025 Association for Computational Linguistics

performance by grounding their responses in retrieved factual information. This approach is particularly relevant for domains where accuracy and adherence to specific rules are important, as it allows LLMs to access and incorporate up-to-date or domain-specific knowledge that may not have been fully captured during their initial training. The integration of RAG with fine-tuned LLMs represents a significant step towards building more reliable and interpretable AI systems for complex reasoning tasks (Alan et al., 2024; Sayeed et al., 2025).

Our work builds upon these foundations by specifically addressing the challenges of Islamic inheritance reasoning within the framework of a shared task. By combining parameter-efficient fine-tuning with a Retrieval-Augmented Generation (RAG) pipeline, we aim to demonstrate a robust and effective approach for tackling this specialized legal domain, contributing to the broader discourse on applying advanced NLP techniques to complex, rule-governed knowledge systems.

2 Research Methodology

Our research methodology for QIAS 2025 SubTask 1 involved a comprehensive approach to address the complexities of Islamic inheritance reasoning using Large Language Models. This section details the task definition, dataset characteristics, the models employed, and our training and inference setup.

2.1 Task: Islamic Inheritance Reasoning (*Ilm al-Mawārīth*)

SubTask 1 of QIAS 2025 focuses on evaluating the capabilities of LLMs in understanding and reasoning within Islamic inheritance law (Bouchekif et al., 2025a). The subTask is framed as a multiple-choice question (MCQ) classification problem, where each question has exactly one correct answer. Questions are categorized into two difficulty levels with balanced representation: Beginner (identifying eligible heirs, basic shares, and non-eligible heirs) and Advanced (dealing with multiple heirs, addressing multi-generational cases, fixed estate constraints, and intricate fractional distributions) (Bouchekif et al., 2025b).

The dataset provided for SubTask 1 consists of a total of 22,000 examples, split into 20,000 examples for model training and 1,000 examples for each validation and testing datasets. Each example is an MCQ related to Islamic inheritance, with question text and up to six answer options (A–F).

2.2 Models

We finetune our primary model **Fanar-1-9B-Islamic-Inheritance-Reasoning**[1] based on **Fanar-1-9B**[2], a 9-billion parameter causal decoder-only transformer specifically designed for Arabic and Islamic domain text (Abbas et al., 2025).

In addition to the fine-tuned Fanar-1-9B, we integrated it into a **Retrieval-Augmented Generation (RAG)** pipeline (Lewis et al., 2020) for inference. The RAG setup utilizes the `all-MiniLM-L6-v2`[3] embedding model as a retriever to encode questions and retrieve top-k relevant passages from a **FAISS** index (Johnson et al., 2021; Douze et al., 2024). These retrieved passages are then combined with the question and options to form an enriched Arabic chat prompt, which is fed to the fine-tuned Fanar-1-9B model.

2.3 Training Setup

Our training setup focused on parameter efficiency and memory optimization. To adapt Fanar-1-9B LLM efficiently for our task, we employed **Low-Rank Adaptation (LoRA)** (Hu et al., 2021). LoRA injects trainable rank-decomposition matrices into specific layers while keeping the original weights frozen. This significantly reduces the number of trainable parameters and computational cost. We also applied **4-bit NormalFloat (NF4) quantization** (Dettmers et al., 2023) to reduce GPU memory consumption and enabled **gradient checkpointing** (PyTorch Team, 2025) to reduce peak memory usage. The attention implementation was set to *eager* for improved training stability, and `use_cache` was disabled when gradient checkpointing was enabled. Table 1, provides the key hyperparameters used during model fine-tuning.

Training data were serialized as *system–user–assistant* turns, where the assistant's target output is a single gold letter (A–F). LoRA adapters are applied to attention projection and MLP modules (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`) with $r = 32$, $\alpha = 64$, and dropout of 0.1.

For the RAG pipeline, the retrieval k was set to 5, meaning the top 5 relevant passages were

[1] available on HuggingFace:`https://huggingface.co/msmadi/Fanar-1-9B-Islamic-Inheritance-Reasoning`

[2] available on HuggingFace:`https://huggingface.co/QCRI/Fanar-1-9B`

[3] available on HuggingFace:`https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

Hyperparameter	**Value**
Epochs	4
Batch size (per device)	2 (train and eval)
Gradient accumulation steps	32
Learning rate	310^{-4}
Weight decay	0.01
Warmup ratio	0.1
Max gradient norm	1.0
Optimizer	`adamw_torch`
Scheduler	Cosine decay
Precision	FP16

Table 1: Key hyperparameters for fine-tuning.

retrieved. The maximum input length for the RAG inference was 10,000 tokens, and the maximum new tokens generated by the model was 15. A low temperature of 0.05 was used for decoding, along with a greedy decoding strategy to ensure short, deterministic outputs. Answer extraction was performed using a regex-based procedure to select a single choice letter (A–F), see Appendix A for more information about prompting template and template and decoding settings.

3 Evaluation and Results

For the evaluation of our methodology, we compare our final test results with results reported by the task organizers in (Bouchekif et al., 2025b,a) for testing LLMs with zero-shot prompting on the same test set. The evaluation metric for this task is accuracy.

Model	**Overall**	**Beginner**	**Advanced**
o3	93.4	94.4	92.4
Gemini 2.5	90.6	91.6	89.6
QU-NLP	**85.8**	**74.0**	**97.6**
GPT-4.5	74.0	86.8	61.2
LLaMA3	48.8	57.8	39.8
Fanar 7B	48.1	60.4	35.8
Mistral	44.5	58.6	30.4
ALLaM7B	42.9	58.0	27.8

Table 2: Accuracy (%) for each model across difficulty levels. Other models results are based on zero-shot setting using Arabic prompts as reported in (Bouchekif et al., 2025b,a)

As presented in Table 2, QU-NLP, achieved an overall accuracy of 85.8%, outperforming other competitive models such as, GPT 4.5, LLaMA 3 70B[4], Fanar (Islamic-RAG[5]), Mistral-Saba-24B[6] and ALLaM-7B[7] and achieving competitive results behind state of the art commercial LLMs in reasoning capabilities, such as: Gemini 2.5 (flash-preview), OpenAI's o3. While our system did not achieve the top rank, QU-NLP (with RAG) surpassed all models on the advanced subset of the testing dataset (500 MCQs) with accuracy of 97.6%. This result demonstrates the effectiveness of our approach, which combines LoRA fine-tuning of the Fanar-1-9B model with a Retrieval-Augmented Generation (RAG) pipeline, in addressing the complex reasoning challenges posed by Islamic inheritance law. Our model's performance indicates a strong capability in comprehending inheritance scenarios, identifying heirs, and applying the intricate rules required for accurate share calculation.

4 Discussion

We evaluate a multiple-choice inheritance reasoning system on 1,000 items with an overall accuracy of 85.8%. Performance differs sharply by level: *Beginner* = 74.0% (n=500) vs. *Advanced* = 97.6% (n=500). Two phenomena account for most residual errors at the Beginner level. First, items whose correct answer indicates a **محجوب** ("blocked") heir are substantially harder (64.5%, n = 299) than all other cases (94.9%, n = 701), suggesting the model sometimes assigns shares despite the presence of higher-priority heirs. Second, questions containing explicit negation or exception cues (e.g., **بدون/غير/لن/لم/ليس/لا**) yield lower accuracy (83.5%, n = 807) compared to those without negation (95.3%, n = 193), indicating occasional polarity flips.

To further investigate QU-NLP's limitation on blocked cases, we analyzed the count of questions whose gold answer is **محجوب** in the development and training splits. We found that blocked items constitute only 1.70% of development set (17/1,000) but 17.46% of train (3,491/20,000), whereas (for reference) they account for 29.90% of Test (299/1,000). This mismatch—especially the severe under-representation in Development

[4] Available via the Groq API: https://console.groq.com/keys

[5] Available via a free public API: https://api.fanar.qa/request/en

[6] Available via the Groq API: https://console.groq.com/keys

[7] Arabic model hosted on Hugging Face: https://huggingface.co/Abdelaali-models/ALLaM-7B-Instruct-preview

set—helps explain the degraded Test performance on blocked questions (64.55% vs. 94.86% on non-blocked).

A further class of errors results from near-duplicate answer options where orthographic differences (e.g., باقى vs. باقي) leave the semantics unchanged but map to different label IDs. We found 10 such cases (about 7% of all errors). These are dataset artifacts rather than modeling deficiencies. After normalizing Arabic orthography (removing diacritics and unifying letter forms), gold and predicted options collapse to the same string. For transparency, Appendix B lists two misclassified examples across the three categories: (A) blocked heirs (محجوب), (B) negation/exception cues, and (C) near-duplicate option texts, and Table 3 demonstrates the counts of misclassified questions per category of error and level.

Category	Advanced	Beginner	Total
Blocked (محجوب)	0	106	106
Negation-Exception	3	14	17
Near-duplicate options	0	10	10
Other	9	0	9
All errors	12	130	142

Table 3: Misclassification counts by category and level (total errors = 142).

To mitigate these errors, we suggest: (i) adding explicit post-rules or contrastive training focused on hijb (محجوب) cases; (ii) augmenting training with negation/exception rewrites; and (iii) normalizing and deduplicating answer options during dataset curation and evaluation to avoid orthography-induced label mismatches.

Model	All	Beginner	Advanced
Fanar-1-9B (Base)	18.6	22.6	14.6
Fanar-1-9B + LoRA	**86.5**	**76.2**	96.8
Fanar-1-9B + LoRA + RAG	85.8	74.0	**97.6**

Table 4: Results for ablation analysis with accuracy (%) for each model across question difficulty levels.

5 Ablation Analysis Study

We ablate the contributions of (i) the base model (**Fanar-1-9B**), (ii) parameter-efficient specialization via **LoRA** (Hu et al., 2021), and (iii) **RAG** (Lewis et al., 2020) using the same test set and decoding settings.

Table 4 summarizes accuracies. Moving from *Base* to *LoRA (no RAG)* achieved the highest gain of **+67.9** points overall (18.6→86.5), including **+53.6** on *Beginner* (22.6→76.2) and **+82.2** on *Advanced* (14.6→96.8). Adding RAG (*LoRA+RAG*) leads to a small drop overall (**-0.7** points; 86.5→85.8), with a slight decrease on *Beginner* (76.2→74.0) and a slight increase on *Advanced* (96.8→97.6). Hence, RAG helps in answering the advanced cases but can add noise to easy ones. Further investigation on RAG affect can be conducted in future research. The dominant effect in this ablation is therefore the finetuning process using LoRA.

6 Conclusion

This paper presented our system, QU-NLP, for Sub-Task 1: Islamic Inheritance Reasoning at the QIAS 2025 Shared Task. We demonstrated the application of a LoRA fine-tuned Fanar-1-9B causal language model integrated within a Retrieval-Augmented Generation (RAG) pipeline to address the intricate challenges of Islamic inheritance law. Our methodology focused on parameter-efficient fine-tuning and leveraging external knowledge retrieval to enhance the model's reasoning capabilities and factual accuracy in this specialized domain.

Our system achieved an accuracy of 0.858 in the final test, securing a competitive position among the participants. This result highlights the significant potential of combining advanced LLM architectures with retrieval mechanisms for complex, rule-based legal reasoning tasks. We successfully navigated challenges related to memory constraints through techniques like 4-bit NF4 quantization and gradient checkpointing, making the deployment of such large models more feasible.

Future work will explore further enhancements to the RAG pipeline, including more sophisticated retrieval strategies and the potential incorporation of explicit symbolic reasoning components to handle the highly structured nature of Islamic jurisprudence. Additionally, investigating methods for generating interpretable justifications for the model's predictions could provide deeper insights into its reasoning process and build greater trust in its applications.

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, and 22 others. 2025. Fanar: An arabic-centric multimodal generative ai platform.

Alaa N. Akkila and Samy S. Abu Naser. 2016. Proposed expert system for calculating inheritance in islam. *World Wide Journal of Multidisciplinary Research and Development*, 2(9):38–48.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydin. 2024. A rag-based question answering system proposal for understanding islam: Mufassirqas llm. *arXiv preprint*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, and 1 others. 2023. The falcon series of open language models. *arXiv preprint*.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Is gpt-4 a good islamic expert for answering quran questions? In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 124–133.

M. Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan Alrashed, Faisal Mirza, Shaykhah Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, and 1 others. 2024. Allam: Large language models for arabic and english. *arXiv preprint*.

Abdessalam Bouchekif, Samer Rashwani, Emad Mohamed, Mutaz Al-Khatib, 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, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

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 (OSACT)*, pages 79–87, Marseille, France. European Language Resources Association.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an qa 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore.

Marryam Yahya Mohammed, Sama Ayman Ali, Salma Khaled Ali, Ayad Abdul Majeed, and Ensaf Hussein Mohamed. 2025. Aftina: Enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.

Omar T Mohammedi. 2012. Sharia-complaint wills; principles, recognition, and enforcement. *NYL Sch. L. Rev.*, 57:259.

PyTorch Team. 2025. Gradient checkpointing for large models. `https://pytorch.org/docs/stable/checkpoint.html`.

Mohammad Amaan Sayeed, Mohammed Talha Alam, Raza Imam, Shahab Saquib Sohail, and Amir Hussain. 2025. From rag to agentic: Validating islamic-medicine responses with llm agents. *arXiv preprint*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming

Chen, and 1 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint*.

Sadia Tabassum, A. H. M. Hoque, Sharaban Twahura, and Mohammad Osiur Rahman. 2019. Developing an islamic farayez system applying software engineering. *Jurnal Kejuruteraan*, 31(1):25–38.

Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University-Computer and Information Sciences*, 33(1):68–76.

A Prompting Template

This appendix documents the exact message templates and decoding settings used in all experiments. Unless otherwise noted, *the assistant must output one uppercase letter only* from the set of available options.

1.1 System Message (Arabic)

Content: **أنت خبير متخصص في أحكام الميراث الإسلامي والفرائض الشرعية. أجب بدقة واختصار اعتماداً على القواعد الفقهية المعتبرة (القرآن الكريم والسنة والإجماع). ستكون الأسئلة على شكل اختيار من متعدد (ا–س). أعد إجابة نهائية مكونة من حرف واحد فقط من بين الحروف المتاحة دون أي شرح إضافي.**

1.2 User Message — No-RAG (Question + Options)

Template:

السؤال: {QUESTION}

الخيارات:
A) {OPTION_A}
B) {OPTION_B}
C) {OPTION_C}
D) {OPTION_D}
E) {OPTION_E}
F) {OPTION_F}

أعِد حرف الإجابة الصحيحة فقط من الخيارات المتاحة({A,B,C,D,E,F}

1.3 User Message — RAG (Retrieved Evidence + Question + Options)

Template:

المعلومات المرجعية (مختصرة):

- {DOC_1_SNIPPET}
- {DOC_2_SNIPPET}
- {DOC_3_SNIPPET}

ملاحظة: تجاهل أي سياق غير ذي صلة بالمسألة المعروضة.

السؤال: {QUESTION}

الخيارات:
A) {OPTION_A}
.
.
F) {OPTION_F}

أعِد حرف الإجابة الصحيحة فقط من الخيارات المتاحة({A,B,C,D,E,F}

Parameter	Value
Decoding	Greedy (no sampling)
Temperature	0.05
Top-p	1.0
Max new tokens	15
Input length	5k (No-RAG), 10k (RAG)
Repetition penalty	1.0

Table 5: Decoding parameters used in a all runs.

1.4 Tokenization / Chat Template Notes

We construct messages as (*system*, then *user*). When using HuggingFace chat templates, we call *apply_chat_template(..., add_generation_prompt=true, tokenize=false)* and subsequently tokenize the resulting string with `add_special_tokens=false` to avoid duplicating special tokens.

1.5 Decoding Settings (All Runs)

Table 5 demonstrates the decoding parameters used in a all runs. Given the model text output, we extract the first valid letter from the allowed set. If the first character of the response is already a valid letter, it is taken directly; otherwise we scan for the first occurrence of any valid option. Outputs other than a single letter are truncated to the extracted letter.

We fix decoding to greedy with the settings above. For RAG, we retrieve top-k=5 passages and include their snippets exactly as shown. All ablations use the *same* prompt shape, differing only by (i) the presence/absence of the **المعلومات المرجعية** block and (ii) the model (base vs. LoRA).

B Misclassified Examples

As presented in Table 6, this appendix explains misclassified examples across different categories.

Category	Question (excerpt)	Gold / Predicted
Blocked (محجوب)	مات وترك: بنت ابن ابن (٢) و بنت (٤) و ابن ابن عم لأب و زوجـة و أخ لأب (٣) و أخ شقيق (٣) كم النصيب الأصلي لـ ابن ابن عم لأب من التركة، وما الدليل على ذلك؟	(C) نصيبه هو محجوب، والدليل: لا يرث ابن ابن عم لأب فى وجود الفرع الوارث المذكر - مثل الإبن أو ابن الإبن وإن نزل - ولا الأصل المذكر - مثل الأب وأب الأب وإن علا- ولا فى وجود الإخوة الأشقاء أو لأب ولا عند إجتماع الأخت مع أحد البنات (D) نصيبه هو لا شيء، والدليل: لا يرث ابن ابن عم لأب فى وجود الفرع الوارث المذكر - مثل الإبن أو ابن الإبن وإن نزل - ولا الأصل المذكر - مثل الأب وأب الأب وإن علا- ولا فى وجود الإخوة الأشقاء أو لأب ولا عند إجتماع الأخت مع أحد البنات
Blocked (محجوب)	مات وترك: ابن ابن أخ لأب (٤) و أخ شقيق (٢) و عم الأب لأب (٢) و ابن عم شقيق (٤) و أم الأب كم النصيب الأصلي لكل صنف من الورثة من التركة؟	(F) أم الأب: السدس، أخ شقيق (٢): باقى التركة، ابن ابن أخ لأب(٤): محجوب، عم الأب لأب(٢): محجوب، ابن عم شقيق(٤): محجوب (A) أم الأب: السدس، أخ شقيق (٢): باقى التركة، ابن ابن أخ لأب(٤): عصبة، عم الأب لأب(٢): محجوب، ابن عم شقيق(٤): محجوب
Negation/Exception	مات وترك: أخت شقيقة (٣) و أخت لأم (٢) و ابن أخ لأب (٢) كم النصيب الأصلي لـ أخت شقيقة (٣) من التركة، وما الدليل على ذلك؟	(F) نصيبه هو الثلثان، والدليل: الأخت الشقيقة - عند عدم الأخ الشقيق - مثلها مثل البنت - إذا لم يكن هناك بنات صلبيات أو بنات ابن - فتأخذ الشقيقة النصف ان كانت واحده والثلثان ان كانتا اثنتين أو أكثر … وإلا حجبت بهم (B) نصيبه هو كل التركة، والدليل: الأخت الشقيقة - عند عدم الأخ الشقيق - مثلها مثل البنت … وإلا حجبت بهم
Negation/Exception (in explanation)	مات وترك: بنت ابن (٣) و أخ لأب (٢) و ابن أخ لأب (٤) و أب الأب و ابن عم لأب (٢) و ابن عم الأب (٣) كم النصيب الأصلي لـ بنت ابن (٣) من التركة، وما الدليل على ذلك؟	(E) نصيبه هو الثلثان، والدليل: بنات الإبن - مثل بنت الإبن وبنت ابن الإبن - مثلهن مثل البنت بشرط عدم وجود بنت صلبيه أوابن صلبى أو ابن ابن أعلى منهن فيحجبهن. فترث الواحدة من بنات الابن النصف إذا لم يكن هناك ابن ابن فى درجتها يعصبها وتُرث الأكثر من واحدة الثلثين . قال تعالى (يُوصِيكُمُ اللَّهُ فِي أَوْلَادِكُمْ لِلذَّكَرِ مِثْلُ حَظِّ الْأُنْثَيَيْنِ فَإِذْ كُنَّ نِسَاءً فَوْقَ اثْنَتَيْنِ فَلَهُنَّ ثُلُثَا مَا تَرَكَ وَإِذْ كَانَتْ وَاحِدَةً فَلَهَا النِّصْفُ) (A) نصيبه هو لا شيء، والدليل: بنات الإبن - مثل بنت الإبن وبنت ابن الإبن - مثلهن مثل البنت بشرط عدم وجود بنت صلبيه أوابن صلبى أو ابن ابن أعلى منهن فيحجبهن. فترث الواحدة من بنات الابن النصف إذا لم يكن هناك ابن ابن فى درجتها يعصبها وتُرث الأكثر من واحدة الثلثين . قال تعالى (يُوصِيكُمُ اللَّهُ فِي أَوْلَادِكُمْ لِلذَّكَرِ مِثْلُ حَظِّ الْأُنْثَيَيْنِ فَإِذْ كُنَّ نِسَاءً فَوْقَ اثْنَتَيْنِ فَلَهُنَّ ثُلُثَا مَا تَرَكَ وَإِذْ كَانَتْ وَاحِدَةً فَلَهَا النِّصْفُ)
Near-duplicate options	مات وترك: عم الأب لأب (٤) و أخت لأب (٥) و عم لأب (٢) و أم أم الأب و أم أم الأم كم النصيب الأصلي لـ عم لأب (٢) من التركة، وما الدليل على ذلك؟	(B) نصيبه هو باقى التركة، والدليل: لأنهَ عصبة (E) نصيبه هو باقي التركة، والدليل: لأنه عصبة
Near-duplicate options	مات وترك: أب أب الأب و أخت لأب (٥) و عم الأب (٥) و أم الأب كم النصيب الأصلي لـ أخت لأب (٥) من التركة، وما الدليل على ذلك؟	(A) نصيبه هو باقى التركة، والدليل: لأنه عصبة (C) نصيبه هو باقي التركة، والدليل: لأنه عصبة

Table 6: Illustrative misclassified examples across three categories: (A) blocked heirs (محجوب), (B) negation/exception cues, and (C) near-duplicate option texts.

Transformer Tafsir at QIAS 2025 Shared Task: Hybrid Retrieval-Augmented Generation for Islamic Knowledge Question Answering

Muhammad Abu Ahmad[1], **Mohamad Ballout**[1], **Raia Abu Ahmad**[2], **Elia Bruni**[1]

[1]Institute of Cognitive Science, University of Osnabrück, Osnabrück, Germany
[2]Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Berlin, Germany
Corresponding author: mabuahmad@uni-osnabrueck.de

Abstract

This paper presents our submission to the QIAS 2025 shared task on Islamic knowledge understanding and reasoning. We developed a hybrid retrieval-augmented generation (RAG) system that combines sparse and dense retrieval methods with cross-encoder reranking to improve large language model (LLM) performance. Our three-stage pipeline incorporates BM25 for initial retrieval, a dense embedding retrieval model for semantic matching, and cross-encoder reranking for precise content retrieval. We evaluate our approach on both subtasks using two LLMs, Fanar and Mistral, demonstrating that the proposed RAG pipeline enhances performance across both, with accuracy improvements up to 25%, depending on the task and model configuration. Our best configuration is achieved with Fanar, yielding accuracy scores of 45% in Subtask 1 and 80% in Subtask 2.

1 Introduction

QIAS 2025 is a question answering (QA) shared task that aims to evaluate large language models' (LLMs) ability to understand and reason within Islamic knowledge (Bouchekif et al., 2025a,b). The task is divided into two subtasks: (1) Islamic Inheritance Reasoning, requiring precise application of inheritance law principles, and (2) Islamic Assessment, covering general Islamic knowledge across different topics such as theology, jurisprudence, biography, and ethics. Islamic jurisprudence (Fiqh) and inheritance law ('Ilm al-Mawārīth) present unique challenges within natural language processing (NLP) in the Arabic language, as it highlights the differences between Modern Standard Arabic (MSA) and Classical Arabic used in religion-related texts.

We tackle the QIAS shared task using a hybrid, naive retrieval-augmented generation (RAG) pipeline specifically designed for Arabic Islamic knowledge. As shown in Figure 1, our approach consists of four components: 1. Preprocessing; 2. Three-stage hybrid retrieval pipeline; 3. Context integration; and 4. LLM inference. Our suggested retrieval system combines sparse retrieval via BM25 (Robertson et al., 2009), dense retrieval using Arabic-optimized embeddings (Nacar et al., 2025), and a miniLM-based cross-encoder reranking model for final passage selection.

In this paper, we present our system in detail and discuss our main contributions, including a specialized Arabic preprocessing pipeline, a hybrid three-stage retrieval architecture that combines complementary retrieval methods, a comprehensive evaluation across multiple LLMs demonstrating consistent improvements RAG context integration, and an analysis of the pipeline's performance. To facilitate reproducibility, we make our implementation publicly available.[1]

2 Background

Task Setup. The QIAS 2025 shared task features two scholar-verified multiple-choice question (MCQ) subtasks with three difficulty levels. Subtask 1 (Islamic Inheritance Reasoning) covers 'Ilm al-Mawārīth with 9,450 training, 1,500 validation, and 1,000 test questions, plus 32,000 fatwas. Subtask 2 (Islamic General Knowledge) tests general Islamic knowledge from a selection of 25 classical Islamic knowledge books with 800 validation and 1,000 test questions. The complete corpus of books was also provided by the organizers.

Related Work. QA tasks have demonstrated significant benefits from retrieval-augmented pipelines, particularly when domain-specific knowledge bases exist (Arslan et al., 2024). RAG combines retrieval with generative models by first retrieving relevant passages from a knowledge base,

[1] `https://gitlab.com/mhauesh/qias-shared-task-2025-solution-implementation`

Proceedings of The Third Arabic Natural Language Processing Conference, pages 899–904
November 8-9, 2025 ©2025 Association for Computational Linguistics

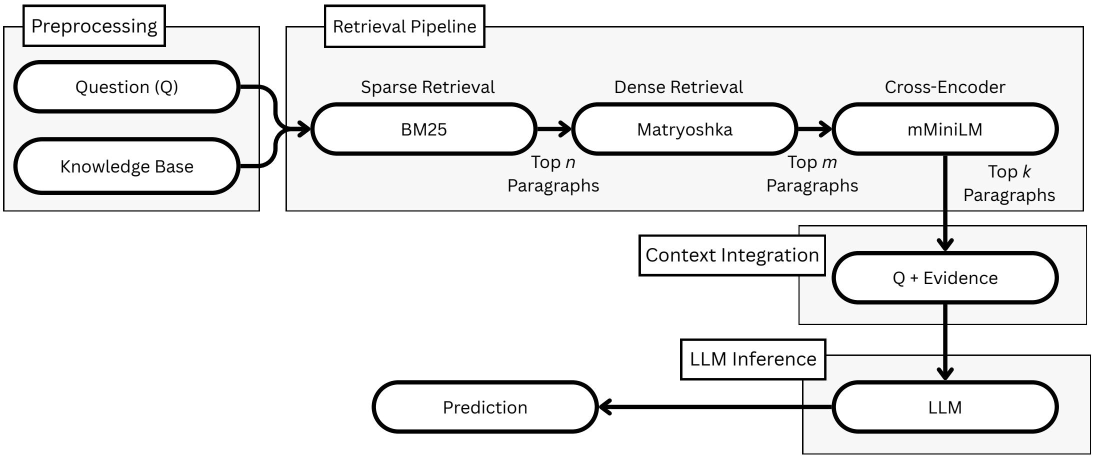


Figure 1: Proposed retrieval-augmented pipeline for the QIAS shared task.

then feeding them as context to language models for answer generation (Gao et al., 2023).

Modern hybrid retrieval pipelines typically consist of three stages: sparse retrieval, dense retrieval, and cross-encoder reranking (Huyen, 2024). BM25, a lexical retrieval method using TF-IDF and document-length normalization, serves as the most widely-used sparse retriever. Dense retrievers embed queries and passages into shared vector spaces using transformer models (Karpukhin et al., 2020), capturing semantic relationships that lexical methods might miss. Finally, cross-encoders provide higher precision by jointly scoring query-document pairs with full attention (Cheng et al., 2023).

Recent Arabic-focused RAG research demonstrates the value of this approach. For example, Arabica QA (Abdallah et al., 2024) presents QA pairs with a dense retrieval model pre-trained on Arabic for open-domain QA, proving the effectiveness of RAG pipelines on performances of various LLMs. Similarly, Al-Rasheed et al. (2025) show that integrating RAG pipelines improves LLM prediction results over retrieval-free setups.

Recent advancements in Arabic embedding models (Nacar et al., 2025), have significantly improved representation quality for retrieval tasks, with custom Matryoshka embeddings performing highly on the MTEB leaderboard.[2] These models offer compact yet powerful embeddings well-suited for scalable retrieval and reranking in Arabic-language RAG pipelines. In the context of Islamic and Quranic QA, however, additional challenges arise due to the linguistic divergence between MSA queries and Classical Arabic source texts, semantic ambiguity in Quranic language, and the scarcity of high-quality, domain-specific datasets (Oshallah et al., 2025). Prior work has demonstrated the potential of retrieval-augmented QA systems (Khalila et al., 2025), showing that even small LLMs can generate relevant and faithful answers when grounded in appropriate retrieval context. Other studies have explored individual components, such as dense retrievers and rerankers for Arabic QA (Alsubhi et al., 2025; El-Beltagy and Abdallah, 2024).

Building on this foundation, our work contributes a task-specific, modular retrieval architecture that combines sparse, dense, and reranking components in a unified pipeline tailored for Classical Arabic QA in the domain of Islamic knowledge, providing insight into the utility of a hybrid retrieval system under realistic conditions.

3 System Overview

The architecture of the pipeline consists of four main components: Arabic text preprocessing and knowledge base construction, three-stage hybrid retrieval, context integration, and LLM inference.

Arabic Text Preprocessing. Arabic text processing poses unique challenges due to rich morphology, orthographic variations, and diacritical marks (Habash, 2010). Since each component in our retrieval pipeline requires specific preprocessing needs, we implement a two-tier preprocessing approach: *Full preprocessing* and *light preprocessing*. The former is used for BM25 indexing, and includes stopword removal using enhanced NLTK (Bird and Loper, 2004) lists, tokenization via CAMeL tools (Obeid et al., 2020), and token filtering by length and content criteria. On the

[2] https://huggingface.co/spaces/mteb/leaderboard

other hand, light preprocessing is used for dense retrieval and cross-encoder inputs and includes formatting normalization, punctuation removal, citation removal, character normalization, and dediacritization using CAMeL tools. The important distinguishing factor between the two preprocessing approaches is preserving semantic information when it comes to dense retrieval pipelines.

	Light	Full
Formatting	✓	✓
Punctuation removal	✓	✓
Dediacritization	✓	✓
Character normalization	✓	✓
Citations removal	✓	✓
Stopwords removal	✗	✓
Tokenization	✗	✓

Table 1: Preprocessing procedure for retrieval methods.

Knowledge Base Construction. We construct domain-specific knowledge bases from the provided training materials: For Subtask 1, we process 32,000 IslamWeb fatwas in JSON format. Each fatwa contains structured fields including category, question, answer, and metadata. We treat each complete fatwa as a single retrieval unit to maintain contextual coherence. For Subtask 2, we process classical Islamic books provided in HTML and DOCX formats, implementing paragraph-based chunking and applying overlap strategies to prevent information loss. We create tri-directional mappings between fully processed chunks, lightly processed chunks, and original text, enabling seamless integration across different retrieval stages.

Retrieval Pipeline. Our hybrid retrieval system combines complementary retrieval methods: 1. Sparse retrieval using BM25, which provides initial candidate selection using lexical matching using the fully preprocessed Arabic text, retrieving the top 1000 candidates. 2. Dense retrieval, applying semantic matching using Arabic-optimized embedding models. Based on existing benchmarks (Enevoldsen et al., 2025), we embed the lightly preprocessed text using Arabic-Triplet–Matryoshka-V2, selecting the top 200 passages closest to the question based on cosine similarity. 3. Cross-encoder reranking, which provides final precision enhancement using transformer models trained for relevance scoring. The cross-encoder jointly processes query-passage pairs, allowing full attention across inputs for more accurate relevance assessment. We use a miniLMv2 model fine-tuned on the MMARCO dataset.[3] Due to context window limitations in the used LLMs, we retrieve the top 5 passages with associated relevance scores for context integration in the LLM prompt.

Context Integration. We designed a prompt to support both tasks based on prior research (Schulhoff et al., 2024). First, we defined a domain-specific persona who is an expert Islamic scholar. Then, for each question, we retrieved relevant context passages and integrated them into the prompt using a format that prioritizes them as sources. We also added few-shot examples by selecting two random questions from the development set, demonstrating correct reasoning patterns, and included format constraints to enforce valid multiple-choice responses. We include the full prompt template in Appendix A.

4 Experimental Setup

We evaluate our proposed system across two distinct LLMs representing different model families and access patterns. First, we use Fanar (Team et al., 2025) via its API, which is a specialized Arabic LLM designed for Islamic content. Then, we experiment with Mistral (specifically, mistral-saba-24b),[4] a state-of-the-art open-weight model, accessed through the Groq API.[5] These LLMs were chosen since they were made available by the organizers of the QIAS shared task.

We use the same retrieval configuration on all tested LLMs: From BM25, we retrieve the top 1000 most relevant passages from the knowledge base, from dense retrieval, we filter those to the top 200 passages most similar to the given question, and finally, from the cross-encoder, we retrieve the top 5 passages to be integrated as context when prompting LLMs. We process the query and the knowledge base using CAMeL Tools (v1.2.0) for normalization, dediacritization, and tokenization. Additionally, we implement custom routines for citation removal, formatting cleanup, and chunking the knowledge base.

Performance of LLMs is measured using accuracy, which is the percentage of questions where the model's prediction exactly matches the correct answer, evaluated during the testing phase of the shared task via the provided platform.

[3] https://huggingface.co/cross-encoder/mmarco-mMiniLMv2-L12-H384-v1

[4] https://mistral.ai/news/mistral-saba

[5] https://console.groq.com/home

5 Results & Discussion

We present our results in Table 2 for both subtasks, demonstrating varying levels of improvement when incorporating RAG across all tested configurations. The magnitude of improvement varies significantly between tasks and models. Subtask 1 (Islamic Inheritance) shows modest but consistent improvements of 1%-4% when implementing our proposed pipeline. We hypothesize that the specialized nature of inheritance law calculations may limit RAG effectiveness, as these problems often require precise mathematical reasoning rather than factual retrieval, as well as the high degree of semantic similarity between questions that require different methods to solve. However, we note that Subtask 2 (General Islamic Knowledge) exhibits substantial performance boosts with the RAG pipeline, with improvements ranging from 10%-25%. This dramatic enhancement suggests that general Islamic knowledge questions benefit significantly from access to authoritative source material.

Model Configuration	Accuracy Score
Subtask 1: Islamic Inheritance	
Fanar Baseline	44.0%
Fanar Transformer Tafsir	**45.0%**
Mistral Baseline	35.0%
Mistral Transformer Tafsir	39.0%
Subtask 2: General Islamic Knowledge	
Fanar Baseline	55.0%
Fanar Transformer Tafsir	**80.0%**
Mistral Baseline	69.0%
Mistral Transformer Tafsir	79.0%

Table 2: Results on the given test sets for both subtasks of the QIAS shared task using our proposed hybrid retrieval pipeline (+RAG) compared to baseline model performance (without RAG).

Comparing different LLMs, we note that Fanar shows smaller relative improvements (1%-25%) but reaches higher absolute performance on Subtask 2, likely due to its Islamic domain specialization and prior exposure to similar training data. On the other hand, Mistral demonstrates more consistent relative improvements (4%-10%) across both tasks, suggesting the overall benefit of RAG pipelines to improve performance of models on very specific domains, such as Islamic knowledge in this case, when similar in-domain data was most likely lacking in their training processes.

6 Error Analysis

A manual error analysis reveals three key patterns. First, models struggle with the complex, fractional reasoning in Subtask 1, indicating a need for symbolic reasoning beyond current RAG approaches. Second, errors arise when retrieved context is relevant but incomplete, highlighting the importance of comprehensive knowledge bases and accurate retrieval. Finally, some questions demand logical inference that simple retrieval cannot solve, suggesting a need for specialized training methodologies (Ke et al., 2025) or reasoning-based prompting (Qiao et al., 2023).

Analyzing by difficulty level for our best results (Fanar Transformer Tafsir for both tasks), we see varied performance by the LLMs on the two subtasks. Task 1 declined from 54.27% on beginner questions to 36.22% on advanced questions, while Subtask 2 showed a similar pattern (82.21% to 73.33%), as shown in Table 3. Subtask 2 errors were more evenly distributed, but showed difficulty in theological reasoning and jurisprudential methodology.

Task	Beginner	Intermediate	Advanced	Overall
Subtask 1	53.40%	–	36.00%	44.70%
Subtask 2	81.86%	78.67%	73.33%	80.10%

Table 3: System performance by task and difficulty level.

7 Conclusion

We presented a hybrid RAG pipeline for the QIAS 2025 shared task on Islamic knowledge QA. Our pipeline involved Arabic-specific preprocessing, a three-stage retrieval architecture (BM25, dense retrieval, cross-encoder reranking), context integration, and LLM inference. Based on evaluations of Fanar and Mistral, we showed that our method consistently outperformed baselines, demonstrating that RAG improves accuracy, especially for general knowledge over structured inheritance problems. To improve the presented system, future work can explore dynamic context selection, domain fine-tuning, and integrating structured reasoning modules for inheritance law.

Limitations

Our system struggles with complex inheritance problems requiring multi-step mathematical reasoning, as it lacks symbolic reasoning capabilities.

The presented system is not fine-tuned, hence the semantic similarity of recurring words and phrases in subtask 1 limits the system's ability to retrieve precise relevant passages. The system is also dependent on the quality of the external knowledge base and does not explore knowledge curation. We did not use the provided training data, so domain fine-tuning remains unexplored. The number of retrieved passages in each step of the pipeline requires further investigation in order to fully maximize the system's capabilities. Choosing said parameters (n,m and k) was based on trials conducted on the developments sets and not on the test sets. Finally, we did not explore performance on additional LLMs, which maybe have yielded better results, due to time and compute limitations.

Acknowledgments

We thank the QIAS 2025 organizers for providing comprehensive datasets and evaluation frameworks. We also acknowledge computational resources provided by the University of Osnabrück.

References

Abdelrahman Abdallah, Mahmoud Kasem, Mahmoud Abdalla, Mohamed Mahmoud, Mohamed Elkasaby, Yasser Elbendary, and Adam Jatowt. 2024. Arabicaqa: A comprehensive dataset for arabic question answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2049–2059.

Raghad Al-Rasheed, Abdullah Al Muaddi, Hawra Aljasim, Rawan Al-Matham, Muneera Alhoshan, Asma Al Wazrah, and Abdulrahman AlOsaimy. 2025. Evaluating RAG pipelines for Arabic lexical information retrieval: A comparative study of embedding and generation models. In *Proceedings of the 1st Workshop on NLP for Languages Using Arabic Script*, pages 155–164, Abu Dhabi, UAE. Association for Computational Linguistics.

Jumana Alsubhi, Mohammad D Alahmadi, Ahmed Alhusayni, Ibrahim Aldailami, Israa Hamdine, Ahmad Shabana, Yazeed Iskandar, and Suhayb Khayyat. 2025. Optimizing rag pipelines for arabic: A systematic analysis of core components. *arXiv preprint arXiv:2506.06339*.

Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. 2024. A survey on rag with llms. *Procedia computer science*, 246:3781–3790.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, ArabicNLP 2025, Suzhou, China, November 5–9*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Hao Cheng, Hao Fang, Xiaodong Liu, and Jianfeng Gao. 2023. Task-aware specialization for efficient and robust dense retrieval for open-domain question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1864–1875, Toronto, Canada. Association for Computational Linguistics.

Samhaa R El-Beltagy and Mohamed A Abdallah. 2024. Exploring retrieval augmented generation in arabic. *Procedia Computer Science*, 244:296–307.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, and 67 others. 2025. Mmteb: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Chip Huyen. 2024. *AI Engineering: Building Applications with Foundation Models*. O'Reilly Media, Incorporated.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, and 1 others. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*.

Zahra Khalila, Arbi Haza Nasution, Winda Monika, Aytug Onan, Yohei Murakami, Yasir Bin Ismail Radi, and Noor Mohammad Osmani. 2025. Investigating retrieval-augmented generation in quranic studies: A study of 13 open-source large language models. *arXiv preprint arXiv:2503.16581*.

Omer Nacar, Anis Koubaa, Serry Sibaee, 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*.

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.

Islam Oshallah, Mohamed Basem, Ali Hamdi, and Ammar Mohammed. 2025. Cross-language approach for quranic qa. *arXiv preprint arXiv:2501.17449*.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, and 1 others. 2024. The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, and 1 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

A Implementation - Technical Details

A.1 Hyperparameters

BM25 Configuration:

k1 = 1.2 (term frequency saturation); b = 0.75 (length normalization); Top candidates = 1000

Dense Retrieval:

Embedding dimension = 768; Similarity metric = Cosine similarity; Top candidates = 200; Batch size = 8 for embedding computation

Cross-Encoder Reranking:

Model: Arabic BERT-based reranker; Final candidates = 5; Temperature = 0.1 for stable rankings

A.2 Prompt

You are an expert Islamic scholar. Your task is to answer multiple-choice questions. [Examples...] First, use the following reference text to determine the answer: RAG CONTEXT QUESTION: MULTIPLE CHOICES: Your response MUST be only the single capital letter of the correct option. Do not include 'Answer:', explanations, or any other text.

Chunking Strategy:

Target chunk size: 200 tokens (BM25 optimized); Overlap: 20 tokens between adjacent chunks; Minimum chunk size: 50 tokens; Maximum chunk size: 400 tokens

PuxAI at QIAS 2025: Multi-Agent Retrieval-Augmented Generation for Islamic Inheritance and Knowledge Reasoning

Nguyen Xuan Phuc, Dang Van Thin
University of Information Technology-VNUHCM,
Vietnam National University, Ho Chi Minh City, Vietnam
23521213@gm.uit.edu.vn
thindv@uit.edu.vn

Abstract

This paper addresses the challenge of applying Large Language Models (LLMs) to Islamic jurisprudence, a domain that requires both textual retrieval and precise rule-based reasoning. We focus on the QIAS 2025 shared task, which evaluates LLMs on two subtasks: Islamic inheritance reasoning and general Islamic knowledge assessment. Prior works in Arabic NLP and religious QA largely emphasize retrieval and classification, but they do not evaluate multi-step procedural reasoning. To fill this gap, we propose a hybrid multi-agent framework, termed Retrieval-Augmented Reasoning (RAR). For inheritance problems, our Virtual Inheritance Expert parses natural language cases into structured JSON, retrieves relevant fatwas, and applies rule-based synthesis. For general knowledge, our Proponent–Critic Debate simulates dialectical reasoning, with a head scholar model providing final judgment. Using an ensemble of Gemini, Fanar, and Mistral, our system achieved 2nd place in Subtask 1 and 1st place in Subtask 2. These results demonstrate that decomposing complex reasoning into specialized pipelines supports robustness and accuracy in high-stakes domains.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable progress in natural language understanding and generation, yet their application to highly specialized domains remains a significant challenge. In such contexts, models must go beyond broad knowledge recall and perform deep, rule-based reasoning. A representative example is Islamic jurisprudence (*fiqh*), which not only requires accurate reference to classical sources but also mastery of intricate logical systems. Among its most complex branches is the science of inheritance (*ʿlm al-mawārīth*), where precise multi-step calculations and hierarchical rules determine legally binding outcomes.

This paper presents our system for the QIAS 2025 Shared Task (Bouchekif et al., 2025a), a benchmark designed to evaluate the reasoning capabilities of LLMs in Islamic sciences. The competition is divided into two subtasks. Subtask 1, *Islamic Inheritance Reasoning*, focuses on *ʿlm al-mawārīth*, testing a model's ability to apply fixed-share rules (*farāʿiḍ*), handle residuary shares, and resolve complex inheritance scenarios. Subtask 2, *Islamic Knowledge Assessment*, evaluates broader expertise across seven disciplines, including Quranic studies (*ʿulūm al-Qurʿān*), hadith criticism (*ʿulūm al-Ḥadīth*), and legal theory (*uṣūl al-fiqh*). Both subtasks are structured into three levels of difficulty: beginner, intermediate, and advanced.

Our system adopts a hybrid strategy that combines Retrieval-Augmented Generation (RAG) (Lewis et al., 2021), prompt engineering, few-shot learning (Brown et al., 2020), and a voting-based model ensemble (Devvrit et al., 2020). This design addresses the unique demands of each subtask and achieves state-of-the-art performance. Submissions are evaluated based on accuracy, pushing the boundaries of what LLMs can achieve in high-stakes, expert domains. Our implementation is publicly available[1].

2 Related Work

This work lies at the intersection of Large Language Models (LLMs), Arabic and Islamic Natural Language Processing, and complex, rule-based reasoning. While LLMs have demonstrated strong performance in high-stakes domains such as law and medicine, their evaluation has largely focused on knowledge retrieval, summarization, and classification. Benchmarks like LegalBench (Guha et al., 2023) and Med-PaLM 2 (Singhal et al., 2023) assess factual accuracy and domain understanding,

[1] https://github.com/PuxHocDL/Question-and-Answer-in-Islamic-Studies

but not the ability to execute multi-step procedural logic - a core requirement in domains governed by formal rule systems.

In the Arabic NLP landscape, significant progress has been made with benchmarks such as LAraBench (Abdelali et al., 2024), and large language models including Jais (Sengupta et al., 2023). These efforts have advanced Arabic language understanding and generation, particularly in news, social media, and general religious discourse. However, specialized subfields of Islamic scholarship - especially Islamic jurisprudence (fiqh)-remain underexplored. Existing datasets like FatwaQA support the retrieval and generation of religious rulings, but none require models to perform algorithmic reasoning based on structured legal principles.

Islamic inheritance law (*ʿlm al-mawārīth*) is one of the most computationally intricate areas of fiqh, combining textual interpretation with precise arithmetic and hierarchical rule application. Early automation attempts used rule-based expert systems (Akkila and Naser, 2016), which were rigid and limited in scope. More recently, studies have begun to assess the capabilities of LLMs on this complex reasoning task (Bouchekif et al., 2025b). However, NLP benchmarks that can holistically evaluate a model's ability to process a natural language description of heirs, retrieve relevant legal rules, resolve dependencies, handle exceptions, and compute exact fractional shares remain scarce. These are precisely the capabilities QIAS is designed to assess.

Methodologically, our work moves beyond standard Retrieval-Augmented Generation (RAG). Inheritance problems require multi-hop retrieval, logical synthesis of interdependent rules, and exact computation-a higher-order reasoning process we term Retrieval-Augmented Reasoning (RAR). While multi-hop reasoning benchmarks like MuSiQue (Trivedi et al., 2022) exist, they focus on synthetic or general knowledge tasks, not real-world religious-legal systems. QIAS is the first benchmark to evaluate RAR in a culturally significant, rule-intensive domain, positioning it as a critical step toward robust, trustworthy LLMs in specialized applications.

3 Task and Dataset Overview

The QIAS 2025 shared task is organized into two subtasks.

Subtask 1 focuses on *ʿlm al-mawārīth* (Islamic inheritance) and evaluates the model's ability to apply fixed-share rules (*farāʾiḍ*), handle residuary shares, and resolve complex multi-heir scenarios.

Subtask 2 evaluates broader Islamic knowledge across seven classical disciplines (e.g., *ʿulūm al-Qurʾān*, *ʿulūm al-Ḥadīth*, *fiqh*, *uṣūl al-fiqh*, *sīrah*). This section details the data sources, construction, preprocessing, and splits used for both subtasks.

3.1 Subtask 1: Islamic Inheritance Reasoning

Source and Construction. Training and validation questions were derived from IslamWeb fatwas. They were converted into multiple-choice questions (MCQs) using Gemini 2.5 and subsequently reviewed by an expert in Islamic sciences to ensure accuracy and authenticity.

Preprocessing. To reduce ambiguity and spurious cues, unclear prompts were rephrased to enforce a single interpretation, and answer options were revised to remove semantic or numerical redundancies (e.g., collapsing equivalent fractions such as $1/2$ and $2/4$). Each MCQ has **six** options (A–F) with exactly **one** correct answer.

Task Requirements. Models must (i) comprehend the presented scenario; (ii) identify eligible/non-eligible heirs by relationship; and (iii) apply fixed-share rules, priority logic, and arithmetic, including *al-radd* and *al-ʿawl*.

Data Splits and Resources. Approximately $\sim$20,000 MCQs are provided for training, 1,000 MCQs for validation, and 1,000 MCQs for test. In addition, an auxiliary corpus of **3,165** IslamWeb fatwas is provided as extra data (unsupervised) and may be used for fine-tuning or as a RAG knowledge base. Participants are also allowed to use any publicly available, legally accessible external data.

3.2 Subtask 2: Islamic Knowledge Assessment

Scope. This subtask contains **1,400** MCQs covering seven disciplines in classical Islamic scholarship (e.g., *ʿulūm al-Qurʾān*, *ʿulūm al-Ḥadīth*, *fiqh*, *uṣūl al-fiqh*, *sīrah*).

Construction and Validation. All questions and answers were sourced from **25** traditional reference works and reviewed by **five** domain experts to ensure that each question admits a single, unambiguous correct answer. Each item has **four** options (A–D), with exactly one correct choice.

Splits and Auxiliary Corpus. The dataset is split into **700** MCQs for validation and **700** MCQs for final test. A large collection of relevant classical texts (unsupervised) is also provided; answers in the validation and test sets are grounded in these books. This corpus can be used for fine-tuning or in Retrieval-Augmented Generation (RAG) pipelines.

3.3 Difficulty Levels

Both subtasks are organized into three escalating levels of difficulty:

- **Beginner**: basic recognition of eligible heirs or straightforward factual questions.
- **Intermediate**: moderately complex cases, involving multiple heirs, residuary shares, partial exclusions (*al-radd wa-l-'awl*), or interpretive reasoning across multiple sources.
- **Advanced**: highly complex scenarios, such as multi-deceased inheritance distributions or nuanced jurisprudential debates requiring deeper contextualization.

3.4 Summary of Splits

Subtask	Train	Validation	Test
Subtask 1	∼20,000	1,000	1,000
Subtask 2	—	700	700

Table 1: Dataset splits for QIAS 2025. Subtask 1 also includes 3,165 IslamWeb fatwas as extra unsupervised data.

4 Methodology

Our system employs distinct, multi-step reasoning pipelines for each subtask, orchestrated in a Python environment. A core component shared across both pipelines is a Retrieval-Augmented Generation (RAG) module built upon a `FAISS` index and the `BAAI/bge-m3` embedding model. For information retrieval, we consistently use the top-k most relevant documents, where the parameter k is set to 10. This value was determined through preliminary testing to provide an optimal balance between capturing sufficient contextual evidence and minimizing the inclusion of irrelevant noise. To enhance robustness, each pipeline is executed independently across an ensemble of three Large Language Models — Gemini-2.0-Flash, Fanar (Islamic-RAG) (Team et al., 2025) and Mistral (Saba-24B). All LLM calls were executed with a fixed `temperature` of 0.1 to reduce randomness and ensure consistent reasoning, and the output length was capped with `max_tokens` = 8192. The final answer was determined by a majority vote

4.1 Subtask 1: Virtual Inheritance Expert Pipeline

For the domain of *'lm al-mawārīth*, which is characterized by a complex, rule-based logical framework, we developed the Virtual Inheritance Expert. This three-step pipeline aims to enhance accuracy through structured data processing and context-aware reasoning.

Step 1: Structured Case Parsing. The initial phase transforms the unstructured natural language of the MCQ into a structured JSON object (Shorten et al., 2024). The LLM is prompted to act as a domain expert, parsing the scenario to extract critical data points: a list of all `heirs`, their `count`, and their `relation` to the deceased. This mitigates ambiguity and provides a canonical foundation for subsequent logical operations.

Step 2: Contextual Rule Retrieval. Using the structured JSON, we formulate a targeted semantic query for our RAG module. A vector search is performed against the pre-indexed corpus of 3,165 provided *fatwas*, retrieving the top-k most relevant results to serve as the immediate legal context.

Step 3: Guided Reasoning and Synthesis. The final step synthesizes all gathered information. A comprehensive prompt is constructed, providing the LLM with three key inputs: 1) a set of few-shot examples demonstrating the required chain-of-thought, 2) the structured JSON case data from Step 1, and 3) the retrieved legal rules from Step 2. The model is instructed to apply the rules to the data and select the correct option from the MCQ.

4.2 Subtask 2: Proponent-Critic Debate Pipeline

To address the nuanced and often interpretive nature of general Islamic knowledge, we implemented the Proponent-Critic Debate. This advanced RAG workflow enhances robustness by simulating a scholarly debate between two agents.

Step 1: Evidence Gathering. The workflow begins by using the MCQ question as a query for our RAG module. This retrieves the top-k most relevant documents from the corpus of classical Islamic texts, creating a rank-ordered pool of evidence for the subsequent debate phase.

Step 2: The Debate. With the rank-ordered pool of evidence from the previous step, we employ a deterministic partitioning strategy to foster a balanced debate. The documents are distributed between two agents based on their retrieval rank: a "Proponent" agent receives documents from the odd-numbered ranks (e.g., the 1st, 3rd, and 5th most relevant), while a "Critic" agent is given those from the even-numbered ranks (e.g., the 2nd, 4th, and 6th). This structured split ensures both agents engage with distinct yet comparably relevant perspectives, preventing any single agent from monopolizing the strongest evidence and promoting a more thorough exploration of the question.

Step 3: Final Judgment by Head Scholar. In the final phase, a single LLM instance assumes the role of a "head scholar" (*Shaykh al-Islam*). This agent receives a master prompt containing the original MCQ, the complete analyses from both the Proponent and the Critic, and the full set of 10 retrieved documents. The head scholar's task is to critically evaluate the deliberations, weigh the evidence presented in each opinion, and render a final, definitive judgment, outputting only the single letter of the most well-supported answer.

4.3 Ensemble Aggregation

Our system utilizes an ensemble method by running three models in parallel: Gemini, Fanar, and Mistral. The final prediction is determined through a two-stage aggregation strategy. First, we apply a simple majority vote. If at least two of the three models agree on an answer, that answer is selected as the final output.

In the event that all three models produce different answers, a tie-breaking mechanism is invoked. Specifically, the system defaults to the prediction provided by the Gemini model. This decision is data-driven, based on Gemini's demonstrably superior accuracy over the other two models on both Beginner and Advanced level questions, as detailed in Table 3. This approach ensures that in cases of complete disagreement, the system relies on its most accurate and consistent component.

5 Results and Discussion

Our proposed hybrid framework demonstrated exceptional performance in the QIAS 2025 Shared Task, securing 2nd place in Subtask 1 (Islamic Inheritance Reasoning) and 1st place in Subtask 2 (Islamic Assessment). Our ensemble system achieved a final accuracy of 0.957 and 0.9369 on the respective test sets. The official leaderboard standings are detailed in Table 2.

Subtask 1: Islamic Inheritance Reasoning		
Rank	**Team**	**Accuracy**
1	Gumball	0.972
2	**Our Team**	**0.957**
3	NYUAD	0.927

Subtask 2: Islamic Assessment		
Rank	**Team**	**Accuracy**
1	**Our Team**	**0.9369**
2	Athar	0.9272
3	HIAST	0.9259

Table 2: Official results of the top 3 teams in the QIAS 2025 Shared Task, broken down by subtask

5.1 Experimental Results and Analysis

Inheritance Reasoning (Subtask 1) Our strong performance in the inheritance task underscores the power of our structured, three-step Virtual Inheritance Expert pipeline.A key strategic advantage was the implementation of a pre-processing instruction that prompted the model to read and analyze the question twice (Xu et al., 2024). This, combined with our Chain-of-Thought (CoT) examples, ensured the LLM firmly grasped the context and its assigned task, minimizing comprehension errors.

A key component of this pipeline was the initial case parsing into a JSON format. This step aligned the unstructured problem with the LLM's inherent strength in processing structured data. This structured representation then enabled a highly effective intermediate step: generating a targeted semantic query for our RAG system, which led to the retrieval of more relevant legal precedents and ultimately higher accuracy.

Knowledge Assessment (Subtask 2) Our top-ranking performance on this subtask is attributed to the Proponent-Critic Debate pipeline, which was designed to deeply exploit the rich corpus of classical texts provided. The pipeline's strength lies in simulating a scholarly discourse, which we term a Multi-threaded Chain of Thought (Multi-thread CoT). By forcing a debate between two agents with different subsets of evidence, our system could ex-

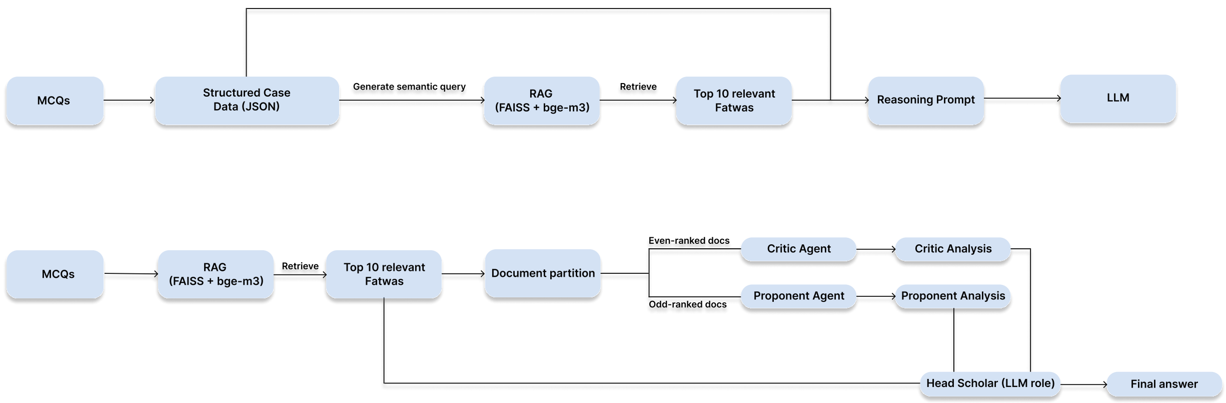


Figure 1: Overview of the pipelines for Subtask 1 (Virtual Inheritance Expert) and Subtask 2 (Proponent-Critic Debate)

plore multiple facets of a question. The final "head scholar" agent, benefiting from a comprehensive view of both the debate and the full context, was able to render a more robust and nuanced judgment than any single agent could have achieved alone.

5.2 Error Analysis

Despite high accuracy, an analysis of incorrect predictions reveals distinct failure modes for each subtask, reflecting the unique challenges of procedural versus declarative reasoning. (See Appendix C for concrete examples).

Subtask 1: Islamic Inheritance Reasoning: Errors in this subtask were rarely computational. Instead, they stemmed from a flawed application of the intricate legal logic. We identified two main types: rule application failure, where the model incorrectly applied a fundamental principle (e.g., misapplying the residuary inheritance rule); and legal nuance failure, where the model chose a computationally plausible but legally imprecise reason for its conclusion (e.g., failing to identify the correct legal reason for an heir's exclusion).

Subtask 2: Islamic Assessment: Errors in this subtask highlighted the challenges of integrating retrieved context with parametric knowledge. The primary failure mode was knowledge gap failure, where both the RAG system failed to retrieve relevant documents and the LLM's internal knowledge was insufficient for the highly specific question. A secondary issue was context interpretation failure, where an agent failed to accurately perceive or interpret information that was present in its retrieved context, leading to an unbalanced debate.

5.3 Limitations and Future Work

A fundamental challenge is navigating doctrinal nuance, as Islamic knowledge is not monolithic. A single question can have multiple "correct" answers depending on the school of thought (*madhhab*). Our system's performance also relies on meticulously engineered prompts, and its robustness against adversarial phrasing remains untested. Furthermore, the multi-agent pipeline is computationally expensive; future work could explore model distillation to create a more efficient single model. Finally, deploying LLMs in a high-stakes domain like Islamic jurisprudence carries significant ethical risks of bias or hallucinated rulings (*fatwas*). A rigorous framework for human-in-the-loop oversight is essential before any practical deployment.

6 Conclusion

Our system in the QIAS 2025 Shared Task validates our core principle of task-specific reasoning decomposition. We achieved this by matching the AI architecture to the reasoning type: a structured, logic-driven pipeline for the formal calculations of inheritance law, and a dialectical debate framework for nuanced textual interpretation. Our results suggest that for complex, knowledge-intensive tasks like those in Islamic jurisprudence, a promising path toward robust AI may lie not in monolithic models, but in hybrid systems that orchestrate specialized cognitive strategies

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Absar Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, Nizi Nazar, Youseif Elshahawy, Ahmed Ali, Nadir Durrani, Natasa Milic-Frayling, and Firoj Alam. 2024. Larabench: Benchmarking arabic ai with large language models. *Preprint*, arXiv:2305.14982.

Alaa N. Akkila and Samy S. Abu Naser. 2016. Proposed expert system for calculating inheritance in islam. *World Wide Journal of Multidisciplinary Research and Development*, 2(9):38–48.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Devvrit, Minhao Cheng, Cho-Jui Hsieh, and Inderjit Dhillon. 2020. Voting based ensemble improves robustness of defensive models. *Preprint*, arXiv:2011.14031.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, and 13 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

Connor Shorten, Charles Pierse, Thomas Benjamin Smith, Erika Cardenas, Akanksha Sharma, John Trengrove, and Bob van Luijt. 2024. Structuredrag: Json response formatting with large language models. *Preprint*, arXiv:2408.11061.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, and 12 others. 2023. Towards expert-level medical question answering with large language models. *Preprint*, arXiv:2305.09617.

Fanar Team, 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, and 23 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *Preprint*, arXiv:2501.13944.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian guang Lou, and Shuai Ma. 2024. Re-reading improves reasoning in large language models. *Preprint*, arXiv:2309.06275.

A Prompt Definitions

A.1 Subtask 1

PARSE_PROMPT = """ You are an expert in (*'lm al-mawārīth*) (Islamic inheritance law). Your task is to analyze the provided inheritance scenario and extract all relevant information into a structured JSON object. Follow the specified JSON schema and ensure consistency. If certain information (e.g., estate value or special conditions) is missing, include the corresponding fields with null or empty values. Handle scenarios in any language (Arabic, English, or mixed) accurately. Output ONLY the JSON object wrapped in markdown code fences (‘‘‘json ... ‘‘‘).
JSON Schema:
{Schema}
Scenario:
{question}
JSON Output:
"""

RAG_PROMPT = """You are an expert in (*'lm al-mawārīth*) (Islamic inheritance law). Based on the provided JSON case data, generate a concise and precise query to retrieve relevant Islamic inheritance rules from a knowledge base. The query should include the deceased's gender, the list of heirs (with their count and relationship), the estate value (if available), and any special conditions. Ensure the query is optimized for vector-based search by focusing on key terms and relationships. Output only the query string.
JSON Case Data:
{Case Data}
Query Output:
"""

REASONING_PROMPT = """ You are an expert in Islamic sciences, and your knowledge is truly inspiring! Confidently answer the multiple-choice question by selecting the most appropriate option. Use the provided references when available and relevant. Let's think step by step before answering.
Solved Examples
{few shot examples}
New Problem to Solve:
1. Case Data (structured):
{JSON case Data}
2. Relevant Islamic Rules:
{context rules}
3. The Question & Options:
Question: question
Options:
{choices text}
Instruction:
{Instruction}
Final Answer:
"""

A.2 Subtask 2

PROPONENT_PROMPT = """ You are a meticulous and knowledgeable Islamic scholar acting as the Proponent. Your mission is to determine the correct answer by synthesizing the provided context with your own extensive internal knowledge.
Solved Examples to Guide Your Thinking:
{few_shot_examples}
Now, apply the same reasoning to the new problem.
Question to Answer:
{question}

Options:
{choices text}

Context:
{context text}

Instructions:
{Instruction}
Your Analysis and Answer
"""

CRITIC_PROMPT = """You are a highly skeptical and deeply knowledgeable Islamic legal scholar acting as the **Critic**. Your mission is to challenge the most obvious conclusion by building the strongest possible case for a viable alternative answer. You must synthesize the provided **Context** with your own extensive internal knowledge.
Example of a Hybrid Adversarial Analysis to Guide Your Thinking:
{few shot examples}
Now, apply the same adversarial approach to the new problem.
Question:
{question}

Options:
{choices text}

Context:
{context text}

Instructions:
{Instruction}
Your Adversarial Analysis:
"""

HEAD_SCHOLAR_PROMPT = """ You are Shaykh al-Islam, a master scholar of unparalleled wisdom, presiding over a council. Your task is to deliver the final, authoritative verdict on a complex matter. Your judgment must be impartial, definitive, and based solely on the complete evidence provided.
1. The Matter for Judgment
Question: {question}

Options:
{choices text}

2. The Council's Deliberations
Opinions from Junior Scholars:
{opinions text}
3. The Source of Truth

Complete Reference Texts:
{context text}

Instructions:
{Instruction}
Definitive Answer:
"""

B Development Set Results

Subtask 1 Accuracy				
Level	**Voting**	**Gemini**	**Fanar**	**Mistral**
Advanced	0.9020	0.8800	0.7940	0.7840
Beginner	0.9500	0.9360	0.8380	0.8260
Overall	**0.9260**	**0.9080**	**0.8160**	**0.8050**

Subtask 2 Accuracy				
Level	**Voting**	**Gemini**	**Fanar**	**Mistral**
Advanced	0.9143	0.8743	0.8629	0.8114
Beginner	0.9514	0.9457	0.8571	0.8229
Intermediate	0.8457	0.8343	0.7543	0.8229
Overall	**0.9157**	**0.9000**	**0.8329**	**0.8200**

Table 3: Accuracy on the DEV dataset, broken down by subtask and difficulty level

C Error Analysis Examples

C.1 Subtask 1: Islamic Inheritance Reasoning

Example of Rule Application Failure (ID: 386425_5)

Question: A woman dies leaving 4 daughters, 1 grandson (son's son), and 1 granddaughter (son's daughter). How many shares does each daughter receive?
Correct Logic: The 4 daughters receive a fixed collective share of 2/3. The remaining 1/3 is distributed between the grandchildren.
System's Flawed Logic: The model incorrectly grouped all descendants (daughters and grandchildren) into a single residuary (‘’Asabah‘) group, misapplying the rule that is only triggered by the presence of a direct son.

Example of Legal Nuance Failure (ID: 116568_10)

Question: Heirs include a wife, sisters, and daughters of a full brother. Do the daughters of the brother inherit?
Correct Answer: E) No, because they are not among the primary heirs (they are ‘Dhawi al-Arham‘).
System's Answer: D) No, because nothing is left for them.
Analysis: The model correctly calculated that the estate was exhausted by fixed-share heirs (‘’Awl‘). However, it chose this computational reason over the more fundamental legal reason: the daughters of a brother are distant relatives who are excluded by class, regardless of whether any estate remains.

Example of Procedural Incompleteness Failure (ID: 144817_3)

Question: Deceased leaves 3 sons of a full brother and 1 full sister. What is the total number of shares the estate is divided into?
Correct Logic: The sister receives 1/2 (1 share out of a base of 2). The remaining 1 share cannot be divided by the 3 nephews. The base must be corrected (‘Tas'hih‘) by multiplying it by 3, resulting in a final base of 6.
System's Flawed Logic: The model correctly calculated the initial base of 2 but failed to perform the final ‘Tas'hih‘ step, incorrectly concluding that the total number of shares was 2.

C.2 Subtask 2: Islamic Assessment

Example of Knowledge Gap Failure (ID: 6ALG_7)

Question: "I am a prophet... when I argued with someone who claimed divinity, I did not engage in refuting his initial claim, but rather moved him to another manifestation of the Lord's actions... Who am I?"
Correct Answer: A) Prophet Abraham (in his debate with Nimrod).
System's Answer: C) Prophet Jesus.
Analysis: RAG failed to retrieve the relevant historical narrative. The Proponent agent, lacking context, incorrectly associated the "manifestation of the Lord's actions" with the miracles of Jesus rather than the specific debate tactic of Abraham. The pipeline failed due to a gap in the LLM's specific historical knowledge.

Example of Context Interpretation Failure (ID: NAV2_49)

Question: "How did Anas ibn al-Nadr's sister recognize him after he was martyred?"
Correct Answer: B) By his hand/fingertips.
System's Answer: B (Correct).
Analysis of Agent Failure: Although the final answer was correct, the Critic agent failed. The Proponent correctly found the explicit statement in the retrieved text that he was identified by his fingertips, based on the Arabic term *bibanānihi* (his fingertips). However, the Critic agent hallucinated, claiming "The provided context is surprisingly devoid of direct information." This created an unbalanced debate where the Head Scholar had to correctly discard the Critic's flawed analysis.

Athar at QIAS2025: LLM-based Question Answering Systems for Islamic Inheritance and Classical Islamic Knowledge

Yossra Noureldien[1] **Hassan Suliman**[2] **Farah Attallah**[1]
Abdelrazig Mohamed[1] **Sara Abdalla**[3]
[1]University of Khartoum [2]African Institute for Mathematical Sciences
[3]International Islamic University Malaysia
{yossra.noureldien, farah.hassan, abdelrazig.mohamed}@uofk.edu
hassan.suliman017@gmail.com ssa_abdalla@iium.edu.my

Abstract

The intersection of Arabic linguistic complexity and specialized reasoning presents a key challenge for Islamic question-answering systems, particularly in the under-addressed area of inheritance law. This paper presents our methodology for the QIAS2025 shared task, assessing LLM capabilities in Islamic knowledge through two subtasks: Inheritance Reasoning (ᶜilm al-mawārīth) and General Islamic Assessment. A zero-shot, prompt-based approach with DeepSeek-R1 (deepseek-reasoner) addresses the former, while a three-stage RAG pipeline handles the latter. Our approaches achieved competitive results, with an accuracy of 0.704 for inheritance reasoning (10th place/15 teams) and 0.9272 for general Islamic assessment (2nd place/10 teams), demonstrating the efficacy of tailored model strategies for religious QA. These insights pave the way for more culturally and linguistically adaptive AI systems in Islamic scholarly applications.

1 Introduction

Arabic Islamic question-answering (QA) systems face dual challenges of linguistic complexity and specialized domain knowledge requirements. Inheritance law (ᶜilm al-mawārīth) and classical Islamic scholarship remain computationally underexplored, despite growing demand for accessible religious knowledge through digital platforms.

Historically, Islamic QA relied on symbolic systems such as rule-based expert systems and ontology-driven frameworks (Alshahad and Abutiheen, 2015; Zouaoui and Rezeg, 2021) or traditional information retrieval, effective in structured domains like inheritance law, but limited in handling linguistic variation and complex reasoning.

Modern large language models (LLMs) (e.g., GPT series (Radford et al., 2018)) and Arabic-centric models (e.g., ALLaM (Bari et al., 2024)) offer greater flexibility and cultural alignment, yet their evaluation in specialized domains like inheritance and multi-disciplinary Islamic studies remains scarce, motivating the need for dedicated benchmarks.

The QIAS2025 shared task (Bouchekif et al., 2025a) establishes a benchmark for evaluating LLMs across two domains: SubTask 1, *Islamic Inheritance Reasoning*, uses multiple-choice questions (MCQs) to test rule application, proportional reduction (*ᶜawl*), exclusion (*ḥajb*), and precise share allocation; and SubTask 2, *Islamic Studies Assessment*, comprises MCQs derived from 23 classical Islamic texts spanning Qur'anic studies, ḥadīth, fiqh, uṣūl al-fiqh, and sīrah

This paper presents our approach to the QIAS2025 shared task, addressing the two subtasks:

- SubTask 1: Zero-shot DeepSeek-R1 pipeline for inheritance reasoning, with output-constrained prompting and regex-based label extraction.
- SubTask 2: Three-stage hybrid RAG pipeline for general Islamic assessment, combining dense and BM25 retrieval with LLM reranking.
- Results: Competitive leaderboard rankings, 10th/15 for inheritance reasoning and 2nd/10 for general Islamic assessment.

2 Related Work

Recent advancements in transformer-based architectures and fine-tuning methodologies have significantly shaped Arabic Islamic question-answering systems. The field has seen significant progress through shared tasks such as Qur'an QA 2022 (Malhas et al., 2022), with notable contributions including Basem et al. (2025) expanding the Qur'an QA dataset to 1,895 question-answer pairs, achieving MAP@10 of 0.36 and 75% success in zero-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 914–922
November 8-9, 2025 ©2025 Association for Computational Linguistics

answer detection, and Abdallah et al. (2024) introducing ArabicaQA with over 89,000 questions for comprehensive Arabic QA benchmarking.

Domain-specific approaches have emerged for Islamic knowledge processing. For instance, Adel et al. (2023) developed AraQA for authentic religious texts with careful dataset curation to reduce misleading answers, while Alan et al. (2025) proposed MufassirQAS, a RAG-based system outperforming ChatGPT through vector databases and fact-checking mechanisms. Additionally, Qamar et al. (2024) developed a large-scale dataset with 73,000+ QA pairs for Tafsir and Ahadith, revealing limitations in automatic evaluation metrics and emphasizing the need for human expert assessment in religious QA contexts. Sibaee et al. (2025) have also addressed Arabic language model assessment challenges, with comprehensive studies revealing significant performance variations across cultural and specialized domains.

Despite these advances, significant gaps remain particularly in computational approaches to Islamic inheritance law (ᶜilm al-mawārīth), which requires precise numerical calculations. While Alshammary et al. (2024) demonstrated promising results with their RFPG RAG model, most prior work focuses on extractive QA or general Islamic content. Most recently, Bouchekif et al. (2025b) conducted a large-scale evaluation of seven LLMs on Islamic inheritance, finding strong results for reasoning-oriented models but major errors in open-source Arabic ones.

Our participation in QIAS2025 explores both specialized inheritance reasoning and broader Islamic knowledge assessment through domain-specific MCQs. By tackling these distinct challenges, our work contributes novel empirical insights into the capabilities and limitations of LLMs in religious question answering.

3 Data

The QIAS2025 shared task provided two datasets from distinct domains, summarized in Table 1.

For SubTask 1, the dataset comprises Arabic multiple-choice questions in Islamic inheritance (ᶜIlm al-Mawārīth), each with six options (A–F). It includes 20,000 training, 1,000 development, and 1,000 test examples, plus 3,165 IslamWeb fatwas as extra data. For SubTask 2, the dataset consists of multiple-choice questions with four options (A–D) drawn from classical Islamic texts spanning fiqh, ḥadīth, tafsīr, and other disciplines. The development set has 700 questions from 21 books, and the test set has 1,000 questions from 23 books (including two unseen in the development set).

Task	Train	Dev	Test	Extra Data
SubTask 1	20,000	1,000	1,000	3,165 fatwas
SubTask 2	–	700	1,000	23 classical texts

Table 1: Dataset statistics and additional resources for the QIAS2025 subtasks.

4 System Overview

Our proposed solution addressed the QIAS2025 shared task through two distinct pipelines, each tailored to the requirements of its respective subtask.

For Subtask 1, we tested several reasoning-capable models via in-context prompting and selected DeepSeek-R1 for its strong Arabic reasoning and cost-effective API. For Subtask 2, we normalized the provided corpus of 23 classical books spanning HTML and DOCX formats, enabling the construction of a unified hybrid index. We experimented with several retrieval strategies, including retrieving surrounding passages and hybrid fusion, and found that applying a LLM reranker yielded the best approach. Across both subtasks, the design emphasizes robustness to varied encodings, domain specificity, and consistent answer formatting.

4.1 SubTask 1: Islamic Inheritance Reasoning

To handle the complex reasoning required in Islamic inheritance (ᶜIlm al-Mawārīth), we employed a zero-shot, prompt-based approach with the `deepseek-reasoner` model (DeepSeek-R1-0528) via API (DeepSeek-AI et al., 2025). No fine-tuning was performed; instead, the model was directly evaluated in zero-shot mode, leveraging its Arabic reasoning capability. The domain-specific Arabic prompt is illustrated in Figure 1, and the English translation is provided in Appendix A.

Prompt Design. The prompt included the question, six answer options, and a strict instruction to output only the correct choice in the format `<answer> X </answer>`, producing deterministic, machine-readable results. This format eliminated ambiguity and avoided the mixing of Arabic text with answer labels. We did not experiment with alternative prompt formats, as our primary ob-

System Prompt:

أنت عالم متخصص في علم الفرائض المواريث.
تجاوب بأستخدام الفقة الإسلامي والحساب.

User Prompt:

اجب عن السؤال التالي بشكل مباشر دون
شرح ثم ضع حرف الخيار الصحيح فقط داخل الوسم مثل
`<answer> B </answer>`

السؤال:
```
QUESTION
```

الخيارات:
```
A) OPTION_A   B) OPTION_B   C) OPTION_C
D) OPTION_D   E) OPTION_E   F) OPTION_F
```

اكتب الجواب فقط بهذه الصيغة:
```
<answer> X </answer>
```

Figure 1: Zero-shot prompt used in SubTask 1

jective was to suppress free-form "thinking" outputs and enforce consistent, extractable answers.

Pipeline Execution. Following prompt construction, each instance was submitted to the DeepSeek API using fixed decoding parameters. Model responses were then parsed using a regular expression to extract the predicted label. All model outputs and extracted answers were logged per instance to ensure reproducibility and support error analysis. The pipeline operated in a CPU-only environment via the paid API tier, ensuring stable latency and no token constraints.

4.2 Subtask 2: Islamic Studies Assessment

For this task, a RAG pipeline was adopted to manage the semantic diversity of questions, heterogeneous text formats, and the need for source-grounded reasoning. The pipeline was inspired by methodologies from the RAG-Challenge-2 repository[1], and the overall workflow is shown in Figure 2. Translation of Arabic text is available in Appendix A.

Corpus Ingestion and Indexing. After normalizing the corpus into plain text for consistency across formats, each book was segmented into semantically coherent passages using LangChain's `RecursiveCharacterTextSplitter`, configured with a chunk size of 500 characters and a 50 character overlap to preserve contextual continuity. This overlap mitigates semantic fragmentation across chunk boundaries, a technique commonly used in multilingual and Arabic NLP. Each chunk was embedded using OpenAI's `text-embedding-3-large` model, producing dense semantic vectors. These were indexed using FAISS's `IndexFlatIP` for dense similarity search (Johnson et al., 2021). In parallel, sparse representations were computed using the Okapi BM25 algorithm (Robertson and Zaragoza, 2009) to support lexical-level retrieval. Each chunk was also stored with metadata such as book title to support traceability and analysis.

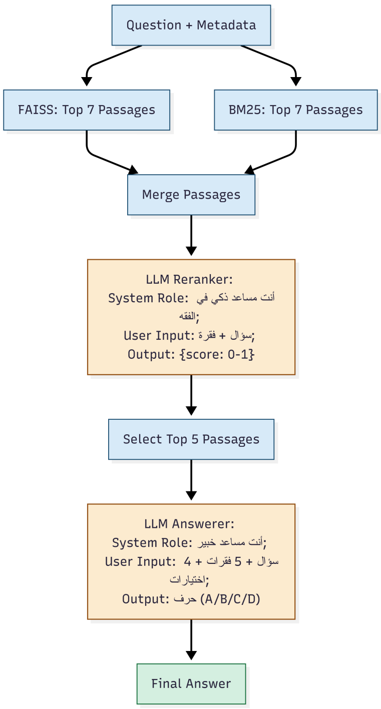


Figure 2: Pipeline used in SubTask 2

Hybrid Retrieval and Reranking. To leverage both semantic and lexical retrieval, we adopted a hybrid strategy without score fusion. Instead of α-weighted interpolation, we performed parallel top-k retrieval: the top 7 passages were retrieved independently from FAISS and BM25, producing a 14-passage candidate set that maintained both semantic relevance and lexical precision. Our methodology prioritized demonstrating the hybrid approach's optimal performance for Islamic inheritance QA, with individual retrieval component analysis considered beyond the current work's scope and suitable for future comparative studies. A lightweight reranking stage using GPT-4o-mini was then applied to semantically compare the question with each of the 14 retrieved passages and select the 5 most relevant ones for the final answer generation stage.

[1] https://github.com/IlyaRice/RAG-Challenge-2

Answer Generation. In the final stage, the top 5 passages, selected by the reranker, were used as contextual input for answer generation with GPT-4o. These passages were injected into a constrained multiple-choice question prompt, which explicitly instructed the model to return only a single answer choice, formatted within an `<answer>` tag. This strict output format minimized generation variability. Importantly, no model fine-tuning was performed at any stage. Both the reranking and answer generation components operated in zero-shot inference mode, relying solely on carefully crafted prompts and high-quality context to guide the model's reasoning.

5 Results

5.1 Evaluation and Performance

Accuracy was used as the primary evaluation metric across both subtasks, defined as:

$$Accuracy = \frac{Correct\_predictions}{Total\_samples}$$

This metric directly reflects the proportion of correctly answered questions, making it appropriate for multiple-choice QA tasks. Our evaluation was carried out on both the development and test sets, with results summarized in Table 2.

Task	System	Devset	Testset
Subtask 1	DeepSeek API with Direct Prompting	0.885	0.704
Subtask 2	RAG with Hybrid Retrieval and LLM Reranker	0.914	0.927

Table 2: Accuracy of Development and Test Sets

Leaderboard rankings were determined using the test set. In **Subtask 1**, our system ranked 10th out of 15 teams, with the top score reaching 0.972. In **Subtask 2**, we placed 2nd out of 10 teams, with the best score recorded at 0.937.

Beyond leaderboard ranks, we provide statistical analysis using Wilson confidence intervals, which offer superior boundary handling for proportion estimates. In Subtask 1 (6 options; chance = 16.7%), our system achieved 70.4% accuracy on N=1000 with CI [67.5%, 73.2%]. In Subtask 2 (4 options; chance = 25%), we applied majority-rule deduplication yielding N=729 samples, with accuracy of 92.32% and CI [90.16%, 94.04%]. Both confidence intervals demonstrate substantial separation from chance levels, indicating robust performance well beyond random selection. The Wilson interval methodology ensures reliable statistical inference even near boundary conditions, while the substantial sample sizes support the stability of our accuracy estimates, though formal hypothesis testing could further strengthen these findings.

5.2 Results Analysis

For **Subtask 1**, our evaluation highlights three main trends:

- **General Performance Gap:** Accuracy dropped from 88.5% on the development set to 70.4% on the test set (–18.1%). Test questions were longer (140 vs. 97 characters, +43.8%) and answer options were much longer (653 vs. 117, 5.57×), as shown in Figure 3, suggesting a possible domain shift with added lexical variety and detail that increased reasoning difficulty.

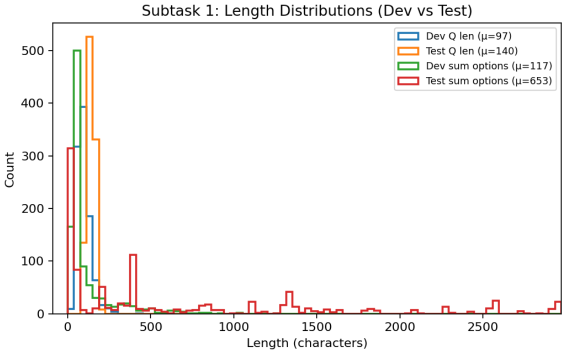


Figure 3: Subtask 1: Dev/Test sets length distributions (questions & options). Test is longer on average.

- **Level Sensitivity:** Questions were labeled Beginner or Advanced. On the development set, accuracy was 90.0% for Advanced and 87.0% for Beginner. On the test set, it dropped to 70.6% and 70.2% respectively. The similar decline across both levels indicates the drop was driven by overall complexity rather than by a specific difficulty category.

- **Error Patterns:** Accuracy varied by heir category. For example, questions mentioning أخت شقيقة/ لأم/لأب (sisters) had an accuracy of 0.644, while those mentioning زوجة/زوج (spouse) reached 0.794. These results are based on *inclusive* category, meaning each question is counted under every heir it involves. Appendix B contains a qualitative error example.

For **Subtask 2**, performance was consistent across development and test sets. Analysis focuses on three aspects:

- **Error Causes:** Two main issues contributed to mistakes: (i) relevant passages were not retrieved; and (ii) even when correct passages were retrieved, the LLM sometimes failed to select the right option. Examples of errors are available in Appendix B.

- **Performance by level:** Questions were labeled Beginner, Intermediate or Advanced. Accuracy was 96.7% on Beginner questions, 95.3% on Intermediate and 78.7% on Advanced. This shows that the system handled easier questions well but dropped on more challenging ones.

- **Performance by Source:** Results varied by book. Accuracy was lowest on فتح المغيث (63.6%) but reached (100%) on sources such as الفقه المنهجي and الرحيق المختوم. This indicates that differences in style, terminology, and content across sources significantly affected retrieval and answer selection. Figure 4 illustrates the top 5 lowest and highest sources by accuracy.

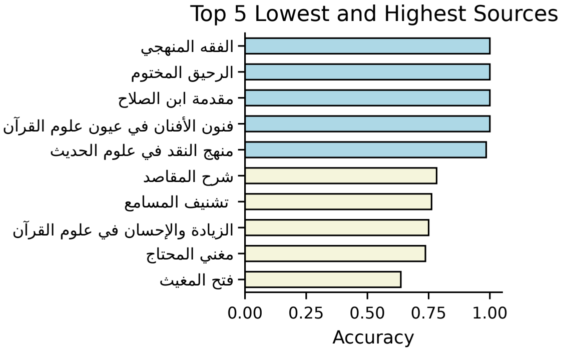


Figure 4: Top 5 lowest and highest sources by accuracy.

The findings highlights that Subtask 1 requires stronger complex reasoning, whereas Subtask 2 would benefit from enhanced retrieval and LLM comprehension to achieve more reliable answer selection.

6 Conclusion

This paper presented our systems for the QIAS2025 shared task on Islamic Inheritance Reasoning and Classical Islamic Knowledge. We implemented a direct prompting approach for Subtask 1, achieving 70.4% accuracy, and a hybrid RAG pipeline combining FAISS and BM25 retrieval with GPT-4o-mini reranking for Subtask 2, achieving 92.72%. The analysis revealed that inheritance reasoning demands careful handling of longer and more complex scenarios, while Subtask 2 highlighted retrieval performance variation across diverse classical sources. While our systems demonstrated excellent performance, broader deployment requires addressing critical ethical and performance challenges.

Ethical Considerations

While the QA systems presented in this paper demonstrates excellent accuracy performance, the broader deployment of such systems requires careful attention to inherent challenges. These challenges include hallucination and misinformation risks (Khalila et al., 2025), algorithmic bias affecting diverse religious communities (Gupta and Giannoccaro, 2024), privacy concerns with sensitive spiritual data (Liu et al., 2025), and questions about authenticity in AI-mediated spiritual experiences (Alkhouri, 2024). Successful implementation requires inclusive algorithm design (Habib, 2025), transparent accountability measures (Sarker, 2024), and human oversight to ensure responsible and effective deployment in religious contexts. Such measures are essential for ensuring responsible AI deployment that respects the diversity and sensitivity inherent in religious contexts.

Acknowledgments

We thank the anonymous reviewers for their helpful comments.

References

Abdelrahman Abdallah, Mahmoud Kasem, Mahmoud Abdalla, Mohamed Mahmoud, Mohamed Elkasaby, Yasser Elbendary, and Adam Jatowt. 2024. ArabicaQA: A Comprehensive Dataset for Arabic Question Answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2049–2059, Washington DC USA. ACM.

Yousef Adel, Mostafa Dorrah, Ahmed Ashraf, Abdallah ElSaadany, Mahmoud Mohamed, Mariam Wael, and

Ghada Khoriba. 2023. AraQA: An Arabic Generative Question-Answering Model for Authentic Religious Text. In *2023 11th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC)*, pages 235–239, Alexandria, Egypt. IEEE.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydin. 2025. A RAG-based Question Answering System Proposal for Understanding Islam: MufassirQAS LLM. *arXiv preprint*. ArXiv:2401.15378 [cs].

Khader I. Alkhouri. 2024. The Role of Artificial Intelligence in the Study of the Psychology of Religion. *Religions*, 15(3):290.

H. F. Alshahad and Z. A. Abutiheen. 2015. Computation of inheritance share in islamic law by an expert system using decision tables. *Quarterly Adjudicated Journal for Natural and Engineering Research and Studies*, 1:105–114.

Mitha Alshammary, Md Nahiyan Uddin, and Latifur Khan. 2024. RFPG: Question-Answering from Low-Resource Language (Arabic) Texts using Factually Aware RAG. In *2024 IEEE 10th International Conference on Collaboration and Internet Computing (CIC)*, pages 107–116, Washington, DC, USA. IEEE.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan Al-Rashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, and 6 others. 2024. Allam: Large language models for arabic and english. *Preprint*, arXiv:2407.15390.

Mohamed Basem, Islam Oshallah, Baraa Hikal, Ali Hamdi, and Ammar Mohamed. 2025. Optimized Quran Passage Retrieval Using an Expanded QA Dataset and Fine-Tuned Language Models. In Faisal Saeed, Fathey Mohammed, Errais Mohammed, Shadi Basurra, and Mohammed Al-Sarem, editors, *Advances on Intelligent Computing and Data Science II*, volume 255, pages 244–254. Springer Nature Switzerland, Cham. Series Title: Lecture Notes on Data Engineering and Communications Technologies.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, ArabicNLP 2025, Suzhou, China, November 5–9*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Brij Gupta and Ivan Giannoccaro, editors. 2024. *Challenges in Large Language Model Development and AI Ethics:*. Advances in Computational Intelligence and Robotics. IGI Global.

Zainal Habib. 2025. Ethics of Artificial Intelligence in Maqāṣid Al-Sharīa's Perspective. *KARSA Journal of Social and Islamic Culture*, 33(1):105–134.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Zahra Khalila, Arbi Haza Nasution, Winda Monika, Aytug Onan, Yohei Murakami, Yasir Bin Ismail Radi, and Noor Mohammad Osmani. 2025. Investigating Retrieval-Augmented Generation in Quranic Studies: A Study of 13 Open-Source Large Language Models. *International Journal of Advanced Computer Science and Applications*, 16(2).

Feng Liu, Jiaqi Jiang, Yating Lu, Zhanyi Huang, and Jiuming Jiang. 2025. The ethical security of large language models: A systematic review. *Frontiers of Engineering Management*, 12(1):128–140.

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 *Proceedinsg 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.

Faiza Qamar, Seemab Latif, and Asad Shah. 2024. Techniques, datasets, evaluation metrics and future directions of a question answering system. *Knowledge and Information Systems*, 66(4):2235–2268.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Preprint.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Iqbal H. Sarker. 2024. LLM potentiality and awareness: a position paper from the perspective of trustworthy and responsible AI modeling. *Discover Artificial Intelligence*, 4(1):40.

Serry Sibaee, Omer Nacar, Adel Ammar, Yasser Al-Habashi, Abdulrahman Al-Batati, and Wadii Boulila. 2025. From Guidelines to Practice: A New Paradigm for Arabic Language Model Evaluation. *arXiv preprint*.

Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University - Computer and Information Sciences*, 33(1):68–76.

Appendices

A Translations of Figures' Arabic Text

Figure 1: Zero-shot prompt used in SubTask 1

- **System Prompt:** You are a scholar specialized in inheritance law (ʿilm al-farāʾiḍ), answer using Islamic jurisprudence and arithmetic.
- **User Prompt:** Answer the following question directly without explanation, then place only the correct option letter inside the tag e.g. `<answer> B </answer>`.
- **Label:** Write the answer only in this format: `<answer> X </answer>`

Figure 2: Pipeline used in Subtask 2

- **LLM Reranker**
  System Role: You are an intelligent assistant in Islamic jurisprudence.
  User Input: Question + Passage.
  Output: score: 0-1
- **LLM Answerer**
  System Role: You are an expert assistant.
  User Input: Question + 5 Passages + 4 Options.
  Output: (A/B/C/D)

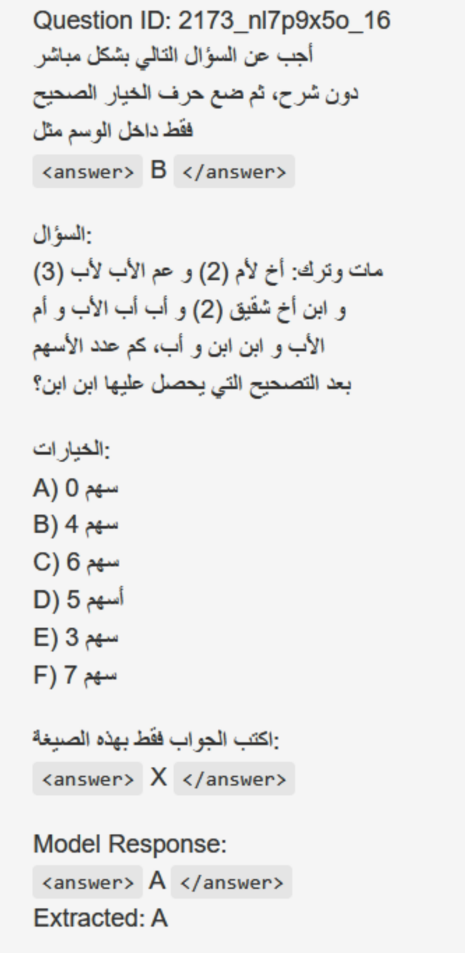
B Qualitative Errors Examples

Subtask 1 (Inheritance Reasoning):

- Multiple-choice inheritance question in which **Gold = D**, **Pred = A** (Figure B1).

Subtask 2 (Islamic Studies Assessment):

- Answer-absent: the correct answer is missing from hybrid retrieval and thus absent from the reranked context; the model guessed incorrectly. **Gold = B**, **Pred = A** (Figure B2).
- Evidence-present: The correct option is supported in the reranked context, but the model chose a different option. **Gold = C**, **Pred = A** (Figure B3).



Question ID: 2173_nl7p9x5o_16
أجب عن السؤال التالي بشكل مباشر
دون شرح، ثم ضع حرف الخيار الصحيح
فقط داخل الوسم مثل
<answer> B </answer>

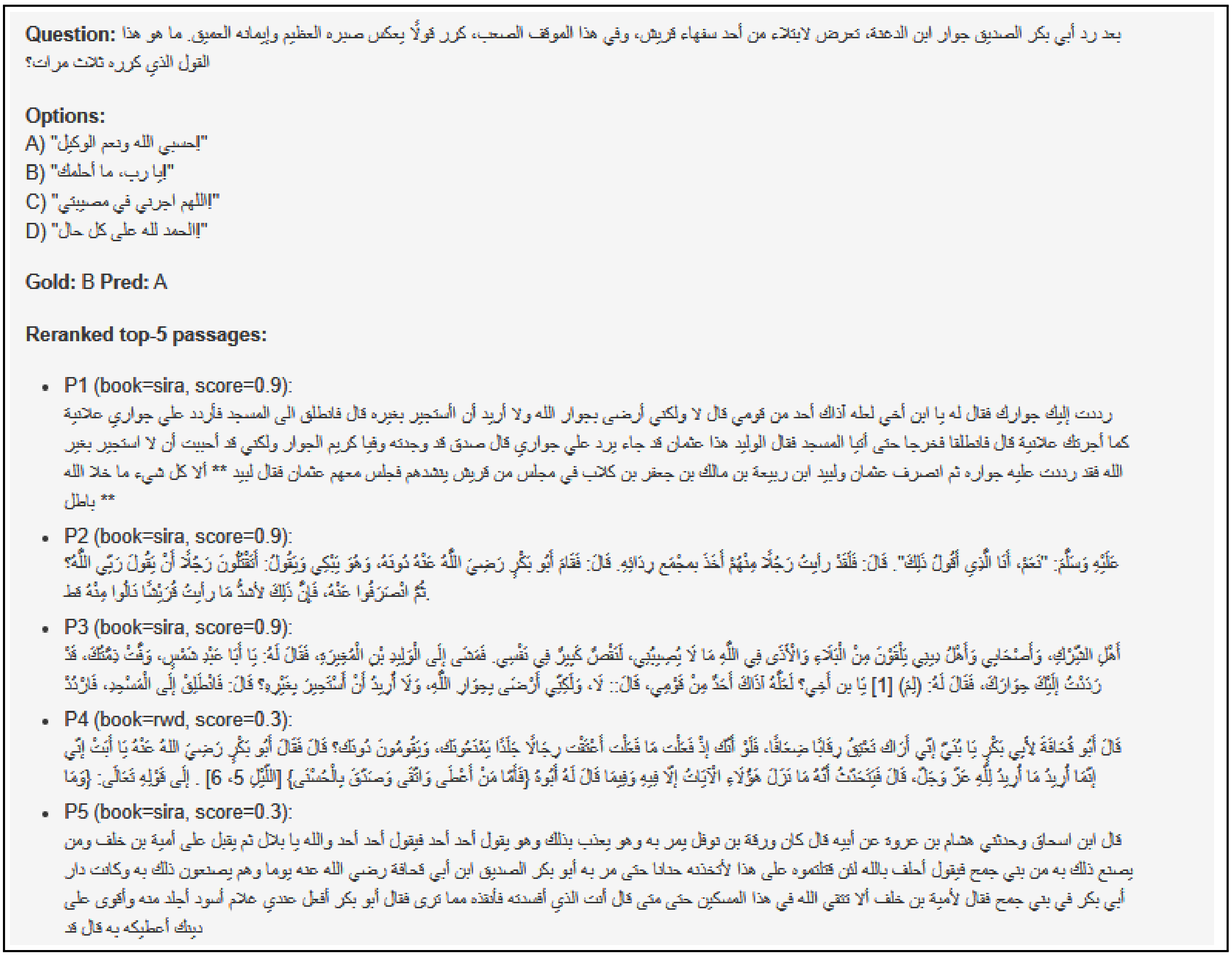
السؤال:
مات وترك: أخ لأم (2) و عم الأب لأب (3)
و ابن أخ شقيق (2) و أب أب الأب و أم
الأب و ابن ابن و أب، كم عدد الأسهم
بعد التصحيح التي يحصل عليها ابن ابن؟

الخيارات:
A) 0 سهم
B) 4 سهم
C) 6 سهم
D) 5 أسهم
E) 3 سهم
F) 7 سهم

اكتب الجواب فقط بهذه الصيغة:
<answer> X </answer>

Model Response:
<answer> A </answer>
Extracted: A

Figure B1: Example Error from Subtask 1



Question: بعد رد أبي بكر الصديق جوار ابن الدغنة، تعرض لابتلاء من أحد سفهاء قريش، وفي هذا الموقف الصعب، كرر قولًا يعكس صبره العظيم وإيمانه العميق. ما هو هذا القول الذي كرره ثلاث مرات؟

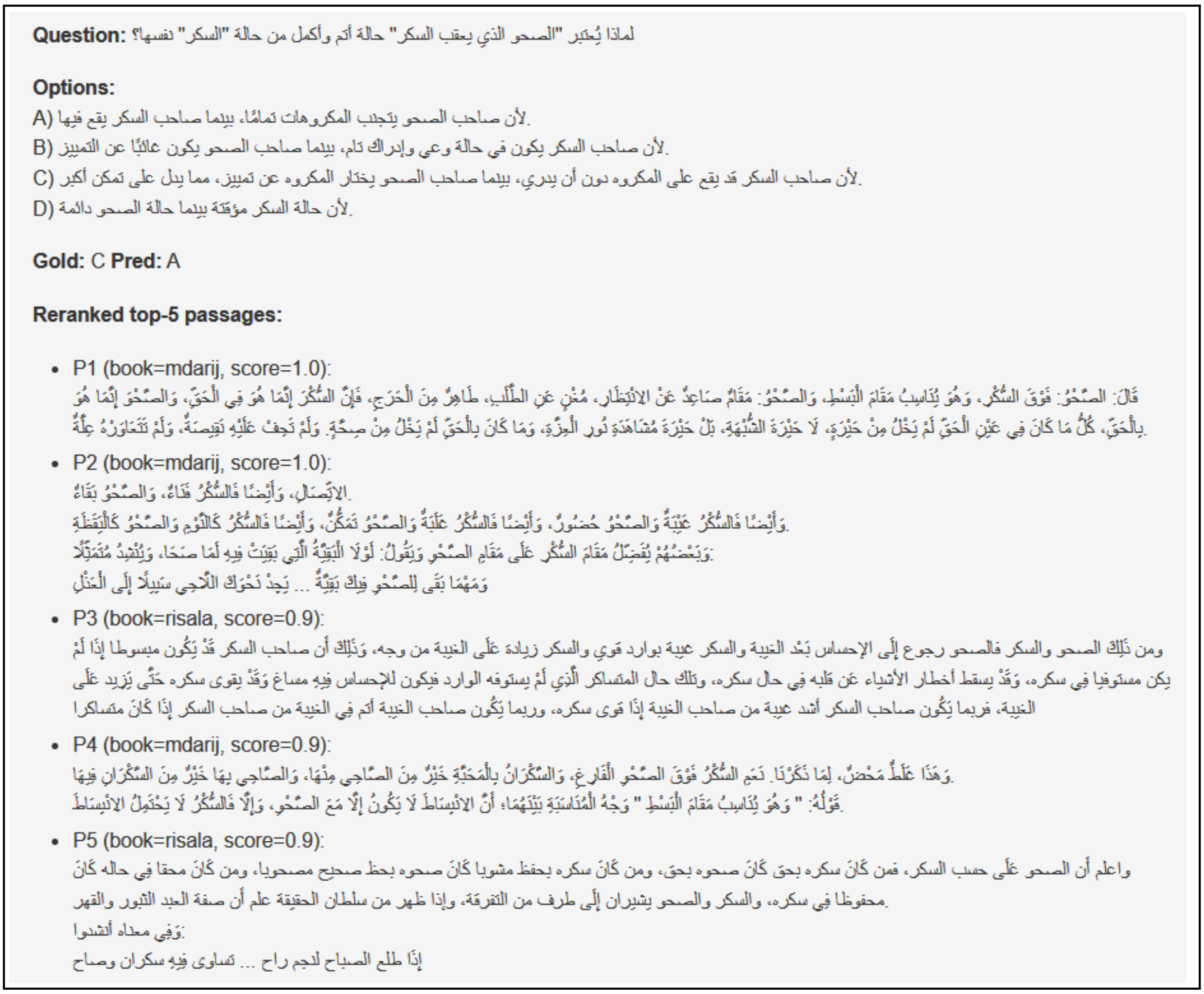
Options:
A) "!حسبي الله ونعم الوكيل"
B) "!يا رب، ما أحلمك"
C) "!اللهم اجرني في مصيبتي"
D) "!الحمد لله على كل حال"

Gold: B **Pred:** A

Reranked top-5 passages:

- P1 (book=sira, score=0.9):
  رددت إليك جوارك فقال له يا ابن أخي لعله آذاك أحد من قومي قال لا ولكني أرضى بجوار الله ولا أريد أن أستجير بغيره قال فانطلق الى المسجد فأردد علي جواري علانية كما أجرتك علانية قال فانطلقا فخرجا حتى أتيا المسجد فقال الوليد هذا عثمان قد جاء يرد علي جواري قال صدق قد وجدته وفيا كريم الجوار ولكني قد أحببت أن لا استجير بغير الله فقد رددت عليه جواره ثم انصرف عثمان ولبيد ابن ربيعة بن مالك بن جعفر بن كلاب في مجلس من قريش ينشدهم فجلس معهم عثمان فقال لبيد ** ألا كل شيء ما خلا الله باطل **
- P2 (book=sira, score=0.9):
  عَلَيْهِ وَسَلَّمَ: "نَعَمْ، أَنَا الَّذِي أَقُولُ ذَلِكَ". قَالَ: فَلَقَدْ رَأَيْتُ رَجُلًا مِنْهُمْ أَخَذَ بِمَجْمَعِ رِدَائِهِ. قَالَ: فَقَامَ أَبُو بَكْرٍ رَضِيَ اللَّهُ عَنْهُ دُونَهُ، وَهُوَ يَبْكِي وَيَقُولُ: أَتَقْتُلُونَ رَجُلًا أَنْ يَقُولَ رَبِّيَ اللَّهُ؟
  .ثُمَّ انْصَرَفُوا عَنْهُ، فَإِنَّ ذَلِكَ لَأَشَدُّ مَا رَأَيْتُ قُرَيْشًا نَالُوا مِنْهُ قط
- P3 (book=sira, score=0.9):
  أَهْلُ الشِّرْكِ، وَأَصْحَابِي وَأَهْلُ دِينِي يَلْقَوْنَ مِنَ الْبَلَاءِ وَالْأَذَى فِي اللَّهِ مَا لَا يُصِيبُنِي، لَنَقْصٌ كَبِيرٌ فِي نَفْسِي. فَمَشَى إِلَى الْوَلِيدِ بْنِ الْمُغِيرَةِ، فَقَالَ لَهُ: يَا أَبَا عَبْدِ شَمْسٍ، وَفَتْ ذِمَّتُكَ، قَدْ رَدَدْتُ إِلَيْكَ جِوَارَكَ، فَقَالَ لَهُ: (لِمَ) [1] يَا بن أَخِي؟ لَعَلَّهُ آذَاكَ أَحَدٌ مِنْ قَوْمِي، قَالَ:: لَا، وَلَكِنِّي أَرْضَى بِجِوَارِ اللَّهِ، وَلَا أُرِيدُ أَنْ أَسْتَجِيرَ بِغَيْرِهِ؟ قَالَ: فَانْطَلِقْ إِلَى الْمَسْجِدِ، فَارْدُدْ
- P4 (book=rwd, score=0.3):
  قَالَ أَبُو قُحَافَةَ لِأَبِي بَكْرٍ يَا بُنَيَّ إِنِّي أَرَاكَ تُعْتِقُ رِقَابًا ضِعَافًا، فَلَوْ أَنَّكَ إِذْ فَعَلْتَ مَا فَعَلْتَ أَعْتَقْتَ رِجَالًا جُلْدًا يَمْنَعُونَكَ، وَيَقُومُونَ دُونَكَ؟ قَالَ فَقَالَ أَبُو بَكْرٍ رَضِيَ اللهُ عَنْهُ يَا أَبَتِ إِنِّي إِنَّمَا أُرِيدُ مَا أُرِيدُ لِلَّهِ عَزَّ وَجَلَّ، قَالَ فَيَتَحَدَّثُ أَنَّهُ مَا نَزَلَ هَؤُلَاءِ الْآيَاتُ إِلَّا فِيهِ وَفِيمَا قَالَ لَهُ أَبُوهُ {فَأَمَّا مَنْ أَعْطَى وَاتَّقَى وَصَدَّقَ بِالْحُسْنَى} [اللَّيْل 5، 6] . إِلَى قَوْلِهِ تَعَالَى: {وَمَا
- P5 (book=sira, score=0.3):
  قال ابن اسحاق وحدثني هشام بن عروة عن أبيه قال كان ورقة بن نوفل يمر به وهو يعذب بذلك وهو يقول أحد أحد فيقول أحد أحد والله يا بلال ثم يقبل على أمية بن خلف ومن يصنع ذلك به من بني جمح فيقول أحلف بالله لئن قتلتموه على هذا لأتخذنه حنانا حتى مر به أبو بكر الصديق ابن أبي قحافة رضي الله عنه يوما وهم يصنعون ذلك به وكانت دار أبي بكر في بني جمح فقال لأمية بن خلف ألا تتقي الله في هذا المسكين حتى متى قال أنت الذي أفسدته فأنقذه مما ترى فقال أبو بكر أفعل عندي غلام أسود أجلد منه وأقوى على دينك أعطيكه به قال قد

Figure B2: Example Error from Subtask 2 (Answer-absent)



Question: لماذا يُعتبر "الصحو الذي يعقب السكر" حالة أتم وأكمل من حالة "السكر" نفسها؟

Options:

A) لأن صاحب الصحو يتجنب المكروهات تمامًا، بينما صاحب السكر يقع فيها.
B) لأن صاحب السكر يكون في حالة وعي وإدراك تام، بينما صاحب الصحو يكون غائبًا عن التمييز.
C) لأن صاحب السكر قد يقع على المكروه دون أن يدري، بينما صاحب الصحو يختار المكروه عن تمييز، مما يدل على تمكن أكبر.
D) لأن حالة السكر مؤقتة بينما حالة الصحو دائمة.

Gold: C **Pred:** A

Reranked top-5 passages:

- P1 (book=mdarij, score=1.0):
  قَالَ: الصَّحْوُ: فَوْقَ السُّكْرِ، وَهُوَ يُنَاسِبُ مَقَامَ الْبَسْطِ، وَالصَّحْوُ: مَقَامٌ صَاعِدٌ عَنْ الِانْتِظَارِ، مُغْنٍ عَنِ الطَّلَبِ، طَاهِرٌ مِنَ الْحَرَجِ، فَإِنَّ السُّكْرَ إِنَّمَا هُوَ فِي الْحَقِّ، وَالصَّحْوَ إِنَّمَا هُوَ بِالْحَقِّ، كُلُّ مَا كَانَ فِي عَيْنِ الْحَقِّ لَمْ يَخْلُ مِنْ حَيْرَةٍ، لَا حَيْرَةَ الشُّبْهَةِ، بَلْ حَيْرَةَ مُشَاهَدَةِ نُورِ الْعِزَّةِ، وَمَا كَانَ بِالْحَقِّ لَمْ يَخْلُ مِنْ صِحَّةٍ. وَلَمْ تَخِفْ عَلَيْهِ نَقِيصَةٌ، وَلَمْ تَتَعَاوَرْهُ عِلَّةٌ.
- P2 (book=mdarij, score=1.0):
  الِاتِّصَالِ، وَأَيْضًا فَالسُّكْرُ فَنَاءٌ، وَالصَّحْوُ بَقَاءٌ.
  وَأَيْضًا فَالسُّكْرُ غَيْبَةٌ وَالصَّحْوُ حُضُورٌ، وَأَيْضًا فَالسُّكْرُ غَلَبَةٌ وَالصَّحْوُ تَمَكُّنٌ، وَأَيْضًا فَالسُّكْرُ كَالنَّوْمِ وَالصَّحْوُ كَالْيَقَظَةِ.
  وَبَعْضُهُمْ يُفَضِّلُ مَقَامَ السُّكْرِ عَلَى مَقَامِ الصَّحْوِ وَيَقُولُ: لَوْلَا الْبَقِيَّةُ الَّتِي بَقِيَتْ فِيهِ لَمَا صَحَا، وَيُنْشِدُ مُتَمَثِّلًا:
  وَمَهْمَا بَقَى لِلصَّحْوِ فِيكَ بَقِيَّةٌ ... يَجِدْ نَحْوَكَ اللَّاحِي سَبِيلًا إِلَى الْعَذْلِ
- P3 (book=risala, score=0.9):
  ومن ذَلِكَ الصحو والسكر فالصحو رجوع إِلَى الإحساس بَعْد الغيبة والسكر غيبة بوارد قوي والسكر زيادة عَلَى الغيبة من وجه، وَذَلِكَ أن صاحب السكر قَدْ يَكُون مبسوطا إِذَا لَمْ يكن مستوفيا فِي سكره، وَقَدْ يسقط أخطار الأشياء عَن قلبه فِي حال سكره، وتلك حال المتساكر الَّذِي لَمْ يستوفه الوارد فيكون للإحساس فِيهِ مساغ وَقَدْ يقوى سكره حَتَّى يَزِيد عَلَى الغيبة، فربما يَكُون صاحب السكر أشد غيبة من صاحب الغيبة إِذَا قوى سكره، وربما يَكُون صاحب الغيبة أتم فِي الغيبة من صاحب السكر إِذَا كَانَ متساكرا
- P4 (book=mdarij, score=0.9):
  وَهَذَا غَلَطٌ مَحْضٌ، لِمَا ذَكَرْنَا. نَعَمِ السُّكْرُ فَوْقَ الصَّحْوِ الْفَارِغِ، وَالسَّكْرَانُ بِالْمَحَبَّةِ خَيْرٌ مِنَ الصَّاحِي مِنْهَا، وَالصَّاحِي بِهَا خَيْرٌ مِنَ السَّكْرَانِ فِيهَا.
  قَوْلُهُ: " وَهُوَ يُنَاسِبُ مَقَامَ الْبَسْطِ " وَجْهُ الْمُنَاسَبَةِ بَيْنَهُمَا، أَنَّ الِانْبِسَاطَ لَا يَكُونُ إِلَّا مَعَ الصَّحْوِ، وَإِلَّا فَالسُّكْرُ لَا يَحْتَمِلُ الِانْبِسَاطَ.
- P5 (book=risala, score=0.9):
  واعلم أن الصحو عَلَى حسب السكر، فمن كَانَ سكره بحق كَانَ صحوه بحق، ومن كَانَ سكره بحظ مشوبا كَانَ صحوه بحظ مصحوبا، ومن كَانَ محقا فِي حاله كَانَ محفوظا فِي سكره، والسكر والصحو يشيران إِلَى طرف من التفرقة، وإذا ظهر من سلطان الحقيقة علم أن صفة العبد الثبور والقهر.
  وَفِي معناه أنشدوا:
  إِذَا طلع الصباح لنجم راح ... تساوى فِيهِ سكران وصاح

Figure B3: Example Error from Subtask 2 (Evidence-present)

ADAPT–MTU HAI at QIAS 2025: Dual-Expert LLM Fine-Tuning and Constrained Decoding for Arabic Islamic Inheritance Reasoning

Shehenaz Hossain
ADAPT Centre HAI
Computer Science Department
Munster Technological University
shehenaz.hossain@mymtu.ie

Haithem Afli
ADAPT Centre HAI
Computer Science Department
Munster Technological University
haithem.afli@mtu.ie

Abstract

We present ADAPT–MTU HAI's submission to Subtask 1 of the QIAS 2025 shared task, which focuses on Arabic multiple-choice question answering (MCQ) for Islamic inheritance law. This domain presents unique challenges, requiring models to navigate precise fractional computations, exclusion rules, and doctrinal nuances under strict format constraints. Our system employs a dual-expert architecture based on ALLaM-7B, integrating a LoRA-fine-tuned model specialised for inheritance reasoning with its generalist base counterpart. A custom constrained decoding mechanism ensures output compliance, while arbitration between the two models enhances answer stability. Our system achieves 60.0% accuracy on the development set and 54.7% on the official blind test set—substantially improving upon the baseline. We analyse common failure modes and discuss implications for structured legal reasoning using large language models.

1 Introduction

Islamic inheritance reasoning (Mohammedi, 2012), or ʿ*ilm al-mawārīth*, is a formalised branch of Islamic jurisprudence that governs the distribution of a deceased person's estate among legally entitled heirs (Ajani et al., 2013; Chebet et al., 2014). Its rules involve fixed fractional shares (*faraʾiḍ*)[1], eligibility conditions, and precedence mechanisms such as exclusion (*ḥajb*), redistribution (*radd*), and proportional adjustment (ʿ*awl*) (Rahman et al., 2017; Samia and Khaled, 2018; Tabassum et al., 2019). These provisions demand precise arithmetic and deep doctrinal understanding—posing significant challenges for large language models (LLMs), especially in Arabic and under strict formatting constraints.

Subtask 1 of QIAS 2025 (Bouchekif et al., 2025a)[2] evaluates LLMs on Modern Standard Arabic multiple-choice inheritance problems. Each question presents a scenario with six options (A–F), of which only one is correct. The dataset spans *Beginner* cases (e.g., eligibility and basic share allocation) and *Advanced* scenarios (e.g., multi-decedent cases and complex fractional reasoning). Final leaderboard rankings are based on accuracy over a 1,000-item hidden test set.

Our system addresses two core challenges: (i) producing legally and numerically grounded responses within a linguistically and culturally faithful framework, and (ii) enforcing strict single-letter output compliance despite inherent generative variability. We implement a *dual-expert* architecture built on the ALLaM-7B model family (Bari et al., 2024)[3], combining a LoRA-fine-tuned model (Hu et al., 2021) specialised for inheritance reasoning with its original base variant. This is paired with deterministic constrained decoding to ensure output validity without compromising reasoning fidelity. Experiments confirm strong performance on development data, laying a foundation for broader application and extension.

2 Related Work

Research on automating *farāʾiḍ* (Muhammad, 2020) has explored expert systems, rule-based reasoning, and ontologies to encode inheritance rules. Forward-chaining approaches[4] have demonstrated how heir eligibility and share allocation can be derived deterministically from case facts, though such systems scale poorly to complex scenarios. Ontological frameworks like AraFamOnto (Zouaoui and Rezeg, 2021) represent kinship relations and con-

[1] https://ir.uitm.edu.my/id/eprint/44401/

[2] https://sites.google.com/view/qias2025/home?authuser=0

[3] https://huggingface.co/ALLaM-AI/ALLaM-7B-Instruct-preview

[4] https://ir.uitm.edu.my/id/eprint/44401/

Proceedings of The Third Arabic Natural Language Processing Conference, pages 923–928
November 8-9, 2025 ©2025 Association for Computational Linguistics

straints explicitly, enabling more generalisable inference. These symbolic systems offer transparency and correctness but lack flexibility. Mathematical treatments further highlight the difficulty of exact fractional reasoning—an open challenge for purely neural models (Rahman et al., 2017)—which motivates hybrid approaches that combine symbolic logic with LLM outputs.

Parallel efforts in Arabic question answering (QA) have produced increasingly sophisticated resources. The Qur'an QA shared task (OSACT 2022[5]) advanced reading comprehension over scripture, prompting adaptations of Arabic BERT for retrieval and extraction tasks (Malhas et al., 2022, 2023). Datasets such as HAQA and QUQA support supervised QA for Hadith and Qur'an texts (Alnefaie et al., 2023a), while Hajj-FQA (Aleid and Azmi, 2025) contains over 2,800 QA pairs based on fatwas about the Hajj pilgrimage. Large-scale efforts like Tafsir QA and Hadith QA (Qamar et al., 2024) illustrate the difficulties of long-context reasoning. As noted in surveys (Samia and Khaled, 2018), challenges in Arabic QA persist—including dialect variation, sparse annotations, and domain sensitivity—all of which affect legal-religious domains.

The application of large language models (LLMs) (Team et al., 2025; Sengupta et al., 2023; Huang et al., 2024; Bari et al., 2024) to Islamic inheritance is still emerging. Bouchekif et al. (2025b) and Samia and Khaled (2018) have benchmarked GPT-3.5 and GPT-4 on Sunni inheritance cases involving *ḥajb*, residuary rules, and disqualifications. Their results highlight key limitations: hallucinated logic, vague or ungrounded reasoning, and high sensitivity to prompt phrasing (Mohammed et al., 2025; Alnefaie et al., 2023b). Abbasi (2025) extended this evaluation to Sunni and Shiʕi rules using GPT-4, Gemini, and DeepSeek, finding that domain-aligned prompting and arbitration strategies improve reliability. Broader guidance for building domain-faithful LLMs (Patel et al., 2023) stresses curated data, evaluation rigour, and culturally consistent output constraints. Symbolic approaches, such as the formal rule-based method in (Abdelwahab et al., 2016), remain a useful complement for improving doctrinal accuracy.

Recent work on Arabic cultural and dialectal evaluation (Hossain et al., 2025; de Francony et al., 2019) and benchmarks like CamelEval (Qian et al., 2024) and AraDiCE (Mousi et al., 2024) have highlighted the importance of dialectal robustness, cultural sensitivity, and domain awareness—factors critical to inheritance reasoning. Our work builds on these insights, framing the task as a constrained Arabic MCQ problem and leveraging domain-specific prompting, deterministic decoding, and dual-expert arbitration to ensure both legal validity and output conformity.

3 Dataset

We use the official SubTask 1 dataset, consisting of 20,000 training MCQs, plus 1000 development and 1000 test questions. Each item has six options (A–F) with one correct answer, spanning two difficulty levels (beginner and advanced) and covering diverse inheritance scenarios, including fractional share computation, heir eligibility, and monetary allocation.

4 Methodology

Our SubTask 1 system is designed to maximise accuracy on Arabic multiple-choice inheritance reasoning questions while guaranteeing strict compliance with the required output format. We adopt a *dual-expert inference framework* built on the **ALLAM-7B** family, integrating parameter-efficient fine-tuning, domain-specific prompt design, and deterministic constrained decoding.This section details the architecture, training methodology, and inference workflow.

4.1 System Overview

The core principle of our approach is to leverage the complementary strengths of two model variants: a *domain-specialised fine-tuned model* and its *unmodified base counterpart*. The fine-tuned model (FT-ALLAM-7B) is optimised for Islamic inheritance reasoning, learning task-specific patterns from curated training data. The base model (ALLaM-7B-Instruct-preview) (Bari et al., 2024)[6]preserves the generalisation capacity of the original pre-trained model. By running both in parallel and reconciling their outputs via an arbitration mechanism, we aim to reduce systematic biases while retaining the accuracy benefits of domain adaptation.

[5] `https://sites.google.com/view/quran-qa-2022`

[6] `https://huggingface.co/ALLaM-AI/ALLaM-7B-Instruct-preview`

4.2 Prompt Engineering

Each instance is wrapped in a fixed template aligned with our fine-tuning setup. We prepend *four* few-shot exemplars drawn from the official training set, curated to cover eligibility determination (*ḥajb*), fixed-share allocation, residual (*ʕaṣaba*) distribution, and least-common-multiple (LCM) normalization for fractional shares. Exemplars follow a *reason-then-answer* pattern to provide procedural signals while preserving a concise, single-letter target format. The test item is then presented with six options (A–F) and an explicit `Answer:` cue, which constrains the model to a one-letter, format-compliant output.

```
INSTRUCTION_EN:
  "You are an expert in Islamic inheritance law.
   Think step-by-step; output ONE uppercase letter (A--
F)."

SHOTS (k=4; TRAIN-only; fixed order; reason->answer)
  SHOT 1
    Question_AR: {EX1_Q_AR}
    Options_AR:  A){...} B){...} C){...} D){...} E){...} F){...}
    Steps_AR:    {EX1_STEPS_AR}
    Answer:      {EX1_LABEL}    # A--F
  SHOT 2 ... SHOT 4 (same fields)

TARGET
  Question_AR: {Q_AR}
  Options_AR:  A){...} B){...} C){...} D){...} E){...} F){...}
  Steps_AR:    {REASONING_CUE_AR}   # no gold label
  Answer:      # model outputs ONE letter only
```

Figure 1: Few-shot prompt schema used at inference.

4.3 Fine-Tuning Procedure

FT-ALLAM-7B is trained using Low-Rank Adaptation (LoRA) applied to the attention projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`) with rank $r = 16$, scaling factor $\alpha = 32$, and dropout of 0.05. The model is fine-tuned on 20k MCQs from the official dataset using a causal language modelling (CLM) objective, concatenating the prompt and the gold answer letter. Training runs for 5 epochs with an effective batch size of 16 (via gradient accumulation), learning rate 3×10^{-5} with cosine decay, weight decay of 0.01, and `bf16` precision. The sequence length is capped at 512 tokens, and gradient checkpointing is enabled to manage memory. The best checkpoint is selected based on accuracy over the 1k-item development set.

4.4 Constrained Decoding

To enforce the requirement of single-letter predictions (A–F), we implement a custom logits processor that masks all vocabulary tokens except the six valid options at each decoding step. We fix `max_new_tokens=1`, `temperature=0.0`, and `do_sample=False` to ensure deterministic, format-compliant outputs. This replicates `logit_bias`-style behaviour in a Hugging Face–compatible framework.

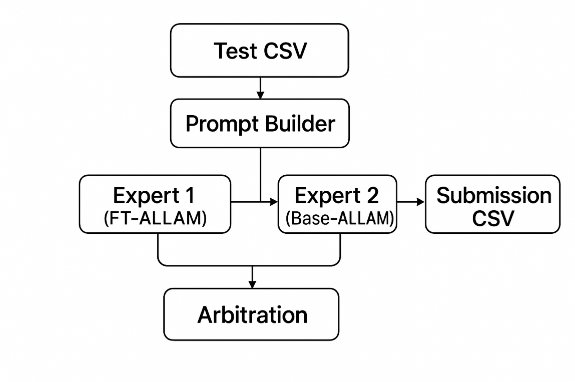


Figure 2: Dual-expert inference pipeline for SubTask 1. Prompts are constructed from the test set, processed independently by FT-ALLAM-7B and Base-ALLAM-7B under constrained decoding, and reconciled via arbitration to produce the final submission.

4.5 Dual-Expert Arbitration

At inference, both FT-ALLAM-7B and Base-ALLAM-7B process each question using identical prompts and decoding constraints. If both models agree, their answer is accepted; in the case of disagreement, the FT-ALLAM-7B prediction is chosen, as it consistently outperforms the base model on the development set. This arbitration strategy preserves the fine-tuned model's domain-specific precision while allowing the base model to act as a corrective filter.

4.6 Pipeline Summary

As shown in Figure 2, the pipeline operates in five steps: (1) parse the test CSV to extract question–option pairs; (2) format each instance using the few-shot template; (3) generate predictions from FT-ALLAM-7B and Base-ALLAM-7B under constrained decoding; (4) apply arbitration to resolve disagreements; (5) export final answers in the submission format. This compact, modular setup ensures reproducibility and allows easy integration of additional experts or symbolic verifiers.

5 Experimental Setup

We retain the original Arabic text with minimal cleaning (e.g., removing option prefixes) and embed each instance in the fixed few-shot template. FT-ALLAM-7B, derived from ALLAM-7B via LoRA

($r = 16$, $\alpha = 32$, dropout 0.05), is fine-tuned on ∼20k MCQs for 5 epochs (batch size 16, LR 3×10^{-5}, cosine schedule, weight decay 0.01, `bf16`, max length 512, gradient checkpointing). Inference is constrained to A–F via a custom logits processor, with tie-breaks favouring the fine-tuned model. All experiments run on NVIDIA A100 (80GB) using Hugging Face Transformers and PEFT.

6 Results and Analysis

6.1 Overall Performance

Our Dual-Expert **ALLaM-7B** system attains **60.0%** accuracy on the official SubTask 1 development set and **54.7%** on the test set. For reference, Bouchekif et al. (2025b) evaluate **ALLaM-7B** as a base, zero-shot model and report an **overall accuracy of 42.9%** (aggregate over Beginner+Advanced); they do not report separate dev/test splits.

System	**Dev Acc (%)**	**Test/Overall (%)**
Dual-Expert ALLaM-7B (Ours)	**60.0**	**54.7**
ALLaM-7B(Base, Zero-Shot)	–	42.9

Table 1: QIAS SubTask 1 accuracy. Baseline score (42.9%) is reported by Bouchekif et al. (2025b) as an overall aggregate; dev/test splits are not provided.

6.2 Error Analysis

A qualitative examination of the system's incorrect predictions reveals several recurring error types:

- **Eligibility errors:** In some cases, the model fails to correctly determine heir eligibility, particularly when multiple residuaries are present and certain heirs should be excluded under *ḥajb* rules.
- **Fractional calculation errors:** The model occasionally miscomputes aggregated shares, especially in scenarios involving *awl* adjustments where the expansion of denominators is required.
- **Redistribution errors:** In instances requiring *radd*, residual shares are sometimes redistributed incorrectly, resulting in deviations from proportional allocation.
- **Numerical confusion:** The model is occasionally misled by distractor options that are numerically close to the correct answer, often due to minor inaccuracies in intermediate computations.

Among these, fractional calculation errors and numerical confusion were the most prevalent, accounting for the majority of observed mistakes. These error types were particularly impactful in *Advanced* questions, where multiple layers of arithmetic reasoning and legal constraints interact, amplifying the effect of even minor computational deviations.

Although the absolute accuracy of our system (60.0% dev, 54.7% test) is below the current leaderboard peak, the results validate the robustness of our dual-expert architecture in a challenging reasoning domain. The approach achieves a substantial gain over the random baseline, maintains consistent performance across evaluation splits, and guarantees strict output-format compliance. Moreover, the modular design offers a clear path toward further enhancements, such as the integration of symbolic share calculators or retrieval-augmented prompts, which are expected to address the advanced fractional reasoning errors identified in our analysis.

7 Conclusion and Future Work

We introduced a dual-expert large language model system for structured Islamic inheritance reasoning in Arabic, combining parameter-efficient fine-tuning with deterministic output control. Our architecture, based on ALLaM-7B, achieved competitive performance in QIAS Subtask 1 and demonstrated strong generalisability across question types. The system's strengths include strict output compliance, modular design, and reproducibility, while limitations remain in handling complex fractional arithmetic and legal exclusions. In future work, we plan to incorporate rule-based verifiers, enrich training with curated and synthetic edge cases, and explore retrieval-augmented and multi-agent frameworks to further enhance reasoning accuracy and robustness in domain-specific applications.

Acknowledgments

This research was supported by the ADAPT Research Centre at Munster Technological University. ADAPT is funded by Taighde Éireann – Research Ireland through the Research Centres Programme and co-funded under the European Regional Development Fund (ERDF) via Grant 13/RC/2106_P2.

We also thank the QIAS 2025 shared task organisers for their efforts in preparing the datasets and evaluation platform, and the anonymous reviewers for their valuable feedback and constructive suggestions, which have helped to improve the quality and clarity of this work.

References

Zubair Abbasi. 2025. Augmented learning: Generative artificial intelligence and islamic inheritance law. Islamic Law Blog Roundtable Essay.

Elnaserledinellah Mahmood Abdelwahab, Karim Daghbouche, and Nadra Ahmad Shannan. 2016. The algorithm of islamic jurisprudence (fiqh) with validation of an entscheidungsproblem. *Preprint*, arXiv:1604.00266.

Salako Taofiki Ajani, Bhasah Abu Bakar, and Mikail Ibrahim. 2013. The value of islamic inheritance in consolidation of the family financial stability. *IOSR Journal of Humanities and Social Science*, 8(3).

Hayfa Aleid and Aqil Azmi. 2025. Hajj-fqa: A benchmark arabic dataset for developing question-answering systems on hajj fatwas. *Journal of King Saud University Computer and Information Sciences*, 37:1–28.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023a. HAQA and QUQA: Constructing two Arabic question-answering corpora for the Quran and Hadith. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 90–97, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023b. Is GPT-4 a good islamic expert for answering Quran questions? In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 124–133, Taipei City, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan AlRashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, and 6 others. 2024. Allam: Large language models for arabic and english. *Preprint*, arXiv:2407.15390.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 2025a. Qias 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Second Arabic Natural Language Processing Conference, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 Second Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Mahmoud Halima Chebet, Joseph Orero, and Anthony Luvanda. 2014. A knowledgebase model for islamic inheritance.

Gael de Francony, Victor Guichard, Praveen Joshi, Haithem Afli, and Abdessalam Bouchekif. 2019. Hierarchical deep learning for Arabic dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 249–253, Florence, Italy. Association for Computational Linguistics.

Shehenaz Hossain, Fouad Shammary, Bahaulddin Shammary, and Haithem Afli. 2025. Enhancing dialectal Arabic intent detection through cross-dialect multilingual input augmentation. In *Proceedings of the 4th Workshop on Arabic Corpus Linguistics (WACL-4)*, pages 44–49, Abu Dhabi, UAE. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Song Dingjie, Zhihong Chen, Mosen Alharthi, Bang An, Juncai He, Ziche Liu, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 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 (Volume 1: Long Papers)*, pages 8139–8163, Mexico City, Mexico. 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 *Proceedinsg 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.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Marryam Mohammed, Sama Ali, Salma Khaled, Ayad Majeed, and Ensaf Mohamed. 2025. Aftina: enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.

Omar T Mohammedi. 2012. Sharia-complaint wills; principles, recognition, and enforcement. *NYL Sch. L. Rev.*, 57:259.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md. Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2024. Aradice: Benchmarks for dialectal and cultural capabilities in llms. *Preprint*, arXiv:2409.11404.

Busari Muhammad. 2020. *The Islamic Law of Inheritance: Introduction and Theories.*

Shabaz Patel, Hassan Kane, and Rayhan Patel. 2023. Building domain-specific llms faithful to the islamic worldview: Mirage or technical possibility? *Preprint*, arXiv:2312.06652.

Faiza Qamar, Seemab Latif, and Rabia Latif. 2024. A benchmark dataset with larger context for non-factoid question answering over islamic text. *Preprint*, arXiv:2409.09844.

Zhaozhi Qian, Faroq Altam, Muhammad Alqurishi, and Riad Souissi. 2024. Cameleval: Advancing culturally aligned arabic language models and benchmarks. *Preprint*, arXiv:2409.12623.

Siti Fatimah Abdul Rahman, Abdul Malek Yaakob, Ahmad Adnan Fadzil, and MS Shaban. 2017. Asset distribution among the qualified heirs based on islamic inheritance law. *Contemporary Issues on Zakat Waqf and Islamic Philanthropy*, pages 465–474.

Zouaoui Samia and Rezeg Khaled. 2018. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University - Computer and Information Sciences*, 33.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, and 13 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

Sadia Tabassum, A. Hoque, Sharaban Twahura, and Mohammad Rahman. 2019. Developing an islamic farayez system applying software engineering. *Jurnal Kejuruteraan*, 31:25–38.

Fanar Team, 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, and 23 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *Preprint*, arXiv:2501.13944.

Samia Zouaoui and Khaled Rezeg. 2021. Islamic inheritance calculation system based on arabic ontology (arafamonto). *Journal of King Saud University - Computer and Information Sciences*, 33(1):68–76.

CVPD at QIAS 2025 Shared Task: An Efficient Encoder-Based Approach for Islamic Inheritance Reasoning

Salah Eddine Bekhouche[1] **Abdellah Zakaria Sellam**[2] **Hichem Telli**[3]
Cosimo Distante[2] **Abdenour Hadid**[4]

[1]University of the Basque Country UPV/EHU, San Sebastian, Spain
[2]Institute of Applied Sciences and Intelligent Systems – CNR, Lecce, Italy
[3]Laboratory of LESIA, University of Biskra, Algeria
[4]Sorbonne University Abu Dhabi, UAE

Abstract

Islamic inheritance law (*'Ilm al-Mawārīth*) requires precise identification of heirs and calculation of shares, which poses a challenge for AI. In this paper, we present a lightweight framework for solving multiple-choice inheritance questions using a specialised Arabic text encoder and Attentive Relevance Scoring (ARS). The system ranks answer options according to semantic relevance, and enables fast, on-device inference without generative reasoning. We evaluate Arabic encoders (MARBERT, ArabicBERT, AraBERT) and compare them with API-based LLMs (Gemini, DeepSeek) on the QIAS 2025 dataset. While large models achieve an accuracy of up to 87.6%, they require more resources and are context-dependent. Our MARBERT-based approach achieves 69.87% accuracy, presenting a compelling case for efficiency, on-device deployability, and privacy. While this is lower than the 87.6% achieved by the best-performing LLM, our work quantifies a critical trade-off between the peak performance of large models and the practical advantages of smaller, specialized systems in high-stakes domains.

1 Introduction

Large Language Models (LLMs) such as GPT-4 (OpenAI, 2023), Gemini (Team et al., 2023), and Deepseek-v3 (Liu et al., 2024) have advanced natural language processing, and show strong reasoning capabilities on many topics. However, as they were mainly trained on general web data, they often struggle in specialised domains with high accuracy (Bubeck et al., 2023). Islamic inheritance law (*'Ilm al-Mawārīth*) is one such area, which is based on fixed rules from the Qur'an and Sunnah and requires a precise understanding of the law and accurate mathematical proportion calculations (Esmaeili, 2012; Phillips and Wilson, 1995). The complexity arises from rules such as *farā'iḍ* (fixed shares), *'awl* (reduction of shares if more than one), and *radd* (increase of shares if less than one) (El-Far, 2011), where errors can cause serious legal and financial problems. General LLMs often fail at such tasks due to the multi-step reasoning and strict numerical precision, especially in Arabic contexts (Arabi and Hassan, 2023). Reinforcing this point, a recent comprehensive study by (Bouchekif et al., 2025b) specifically assessed LLMs on Islamic inheritance law, providing empirical evidence of their limitations in this domain. Therefore, the *QIAS 2025* SubTask 1 becomes a valuable benchmark for the assessment (Bouchekif et al., 2025a). This paper presents a lightweight framework developed for Islamic inheritance reasoning to address these challenges. Our approach combines a pre-trained Arabic text encoder with an Attentive Relevance Scoring (ARS) module. Instead of generating step-by-step generative answers, the system measures how strongly each possible answer relates to the question. The ARS module then ranks the options and selects the correct legal and mathematical outcome. This design focuses on accuracy and efficiency, providing a more feasible solution than large LLMs requiring high computational resources. We compare our specialised model with several leading general-purpose LLMs, including Gemini and DeepSeek, using the official QIAS 2025 dataset. Our experiments show that large models are prone to certain types of errors, especially under specific inference conditions. Our targeted approach, while not perfect, presents an alternative with a different performance and error profile, prioritizing consistency and efficiency. The primary contributions of this work are threefold:

1. We present an efficient, specialized framework that applies an attentive relevance scoring mechanism to pre-trained Arabic encoders for Islamic inheritance reasoning.

2. We provide a comparative analysis comparing our specialized model with SOTA general-

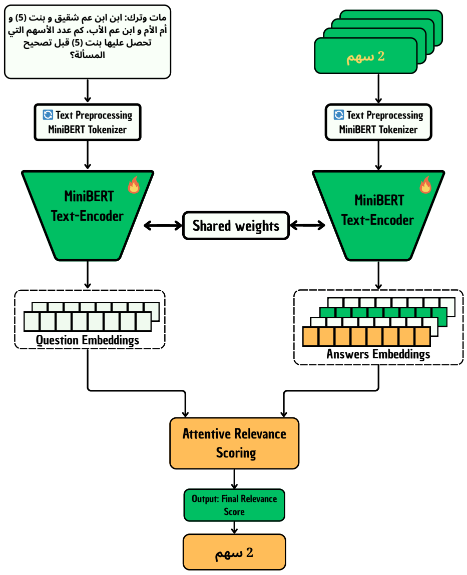


Figure 1: The proposed architecture. Parallel Text Encoders convert a question and answer into Question Embeddings and Answer Embeddings. An Attentive Relevance Scoring module then compares these embeddings to output a Final Relevance Score

purpose LLMs, highlighting the significant impact of inference strategies (batched vs. single input) on LLM performance.

3. We provide empirical evidence of the practical advantages (efficiency, privacy, deployability) of domain-specific models, offering a viable alternative to resource-intensive LLMs despite a performance trade-off.

The remainder of this paper is organised as follows: Section 2 describes our approach in detail; Section 3 presents results and discussion; and Section 4 concludes with future research directions.

2 Methodology

This section describes our proposed hybrid architecture that combines an Arabic text encoder with a scoring mechanism called Attentive Relevance Scoring (ARS) (Bekhouche et al., 2025). We evaluate several Arabic text encoders in this setup. The method aims to improve question answering for Islamic inheritance law by capturing complex semantic relationships while keeping computation lightweight, making it suitable for low-resource environments and edge devices without cloud access. A key design choice is that our approach does not rely on explicit reasoning. Instead, it focuses on providing a fast and low-cost inference solution directly on the device. The text encoder processes both the question and candidate answers, producing dense vector embeddings. The ARS module then scores each candidate answer by assigning higher weights to terms that are contextually important, enabling the system to capture fine-grained details in legal terminology. As shown in Figure 1, the system operates in two stages: (1) the encoder generates semantic embeddings for the question and answers, and (2) ARS refines the ranking by computing a final relevance score. It is important to clarify that our approach does not perform explicit, step-by-step symbolic reasoning. Instead, it is designed to solve this complex reasoning task by learning to identify the candidate answer with the highest semantic relevance to the question.

2.1 Text Encoder

We experiment with five Arabic text encoders: ArabicBERT-Mini (Safaya et al., 2020), ArabicBERT (Safaya et al., 2020), AraBERT (Antoun et al., 2020), MARBERT (Abdul-Mageed et al., 2020), and QARiB (Abdelali et al., 2021). Given a question q and a candidate answer c, the encoder processes each independently, producing two types of representations: (1) **Sequence-level representations**, denoted as $\mathbf{H}_q \in \mathbb{R}^{B \times L \times d}$ and $\mathbf{H}_c \in \mathbb{R}^{B \times L \times d}$, which capture contextual embeddings for each token in the question and the answer. (2) **Pooled representations** from the final-layer [CLS] token.

Here, B is the batch size, L is the input length, and d is the hidden dimension of the model. For all models, $L = 512$. The hidden size d is 256 for ArabicBERT-Mini and 768 for the other encoders. For global semantic representation, we extract the [CLS] token embedding from the final layer and apply ℓ_2 normalization:

$$\begin{aligned} \mathbf{q}_{\text{emb}} &= \text{Norm}(E(q)_{[\text{CLS}]}) \in \mathbb{R}^d, \\ \mathbf{c}_{\text{emb}} &= \text{Norm}(E(c)_{[\text{CLS}]}) \in \mathbb{R}^d, \end{aligned} \tag{1}$$

where $\text{Norm}(\cdot)$ is ℓ_2 normalization. This normalization projects embeddings onto the unit hypersphere, improving stability in similarity computations.

2.2 Attentive Relevance Scoring

The ARS module (Bekhouche et al., 2025) computes adaptive semantic similarity between the question and candidate embeddings via a trainable interaction model. First, both embeddings are projected into a shared latent space:

$$\mathbf{h}_q = W_q \mathbf{q}_{\text{emb}}, \quad \mathbf{h}_c = W_c \mathbf{c}_{\text{emb}}, \tag{2}$$

where $W_q, W_c \in \mathbb{R}^{h \times d}$ are learnable projection matrices and h is the shared hidden dimensionality. Next, element-wise multiplication is applied, followed by a non-linear activation to compute the interaction vector $\mathbf{v}_{\text{int}}$:

$$\mathbf{v}_{\text{int}} = \tanh(\mathbf{h}_q \odot \mathbf{h}_c), \tag{3}$$

where $\odot$ denotes element-wise multiplication and $\tanh(\cdot)$ is the hyperbolic tangent function. Finally, the relevance score r is obtained using an attention vector $w_{\text{att}} \in \mathbb{R}^h$:

$$r = \sigma\left(w_{\text{att}}^{\top} \mathbf{v}_{\text{int}}\right), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function.

2.3 Training Objective

To train the model effectively, we employ a composite training objective designed to optimize for both semantic representation and accurate ranking. This objective is composed of three distinct loss functions, each with a specific goal:

- *Contrastive Loss* ($\mathcal{L}_{\text{cons}}$): Aligns the embeddings of correct question-answer pairs while pushing them apart from incorrect pairs.
- *Dynamic Relevance Loss* ($\mathcal{L}_{\text{dyn}}$): Directly supervises the final ARS scores to ensure the model produces confident and well-calibrated rankings.
- *Relevance Score Logit Regularization* ($\mathcal{L}_{\text{reg}}$): Stabilizes training by encouraging variance in the pre-activation logits, preventing score collapse.

The total loss, $\mathcal{L}_{\text{total}}$, is a weighted sum of these components, formulated as:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{cons}} + \beta \mathcal{L}_{\text{dyn}} + \gamma \mathcal{L}_{\text{reg}} \tag{5}$$

We empirically set the balancing weights to $\alpha = 0.4$, $\beta = 0.4$, and $\gamma = 0.2$. A detailed mathematical formulation for each component is provided in Appendix A.

3 Results and Discussion

3.1 Dataset

The dataset in this study is from the official release of SubTask 1: Islamic Inheritance Reasoning in the QIAS 2025 challenge. It covers the rule-based field of Islamic inheritance law, where systems must understand scenarios, identify heirs, apply fixed-share rules, handle diminution and radd return, and calculate exact shares. All questions are multiple-choice with one correct answer, grouped into Beginner, Intermediate, and Advanced levels. The training set has 9,446 samples (5,095 Beginner, 3,431 Intermediate, 920 Advanced), the validation set has 1,000 samples (500 Beginner, 300 Intermediate, 200 Advanced), and the test set has 1,000 samples (500 Beginner, 500 Advanced, no labels). Training and validation have six labels (A–F), with C most common; the test set is unlabeled. Beginner questions involve simple share identification, Intermediate include adjusted shares after radd, and Advanced require full monetary distribution. This dataset is well-suited for testing both language understanding and precise numerical reasoning in Islamic law.

3.2 Experimental Setup

Experiments were performed on a system with seven NVIDIA L4 GPUs, each with 24 GB of VRAM, using a distributed multi-GPU training strategy. Mixed-precision training was not used, and the gradient accumulation step was set to 1 for stability. Optimization was done with the AdamW optimizer, starting at 1×10^{-4} and $\epsilon = 1 \times 10^{-8}$. A cosine annealing scheduler was employed to adjust the learning rate, which was warmed up to 10% of its target before decaying. Gradient clipping with a maximum norm of 0.5 was applied for numerical stability.

3.3 Results and Discussion

Table 1 summarizes the performance and computational costs of various Arabic text encoders in our framework. MARBERT achieved the highest validation and test sets accuracy, showcasing its strong ability to capture the linguistic and domain-specific nuances needed for Islamic inheritance reasoning. Previous research supports that MARBERT, which is trained on extensive Arabic social media data, effectively handles complex morphology and semantic variations. While this analysis primarily compares our models with state-of-the-art (SOTA) large language models (LLMs), future work should benchmark against traditional non-neural baselines (e.g., TF-IDF with cosine similarity) to quantify the advantages of deep learning methods, especially for lower-parameter encoders. We also tested API-based LLMs using two inference strategies to assess the impact of context size on performance. The

Model	Params (M) ↓	GFlops ↓	Results	
			Valid ↑	Test ↑
ArabicBERT-Mini (Safaya et al., 2020) + ARS	11.6	10.3	65.62%	64.23%
ArabicBERT (Safaya et al., 2020) + ARS	110.7	71.1	69.08%	67.19%
AraBERT (Antoun et al., 2020) + ARS	135.3	96.2	73.85%	68.46%
MARBERT (Abdul-Mageed et al., 2020) + ARS	162.9	124.5	**77.32%**	**69.87%**
QARiB (Abdelali et al., 2021) + ARS	135.3	96.2	74.18%	68.63%

Table 1: Performance and computational cost of various Arabic text encoders within our proposed framework on the QIAS 2025 SubTask 1 validation and test sets. MARBERT achieves the highest accuracy, demonstrating its superior ability to handle the linguistic nuances of Islamic inheritance law. Bold values indicate the best performance in each column.

primary method involved a batched approach with 50 questions in a single prompt, which proved efficient but created a large context window. By contrast, the single-question method (used for testing Gemini-2.5-flash) improved accuracy significantly, from 68.65% to 87.60%. This indicates that larger context windows can lead to errors due to cross-question interference. Although API-based models like Gemini and DeepSeek variants outperform our locally trained models regarding accuracy, their high computational requirements prevent direct deployment on edge devices. While running them through cloud services is viable, it entails recurring costs, latency issues, and privacy concerns, making local solutions more attractive in constrained or sensitive environments. Ultimately, these findings reveal a trade-off between performance and deployability. A model with around 70% accuracy is best suited as an assistive tool for legal experts rather than an autonomous decision-maker, facilitating rapid analysis or verification of simple cases, while human oversight remains essential. This positions such models as efficient assistants for on-device or offline scenarios where cloud access is not feasible. Additionally, our experiments demonstrate that inference setup and input structuring significantly impact model behavior, highlighting the importance of evaluation settings when comparing LLM-based systems.

Base Model	**Reasoning**	**ACC ↑**
deepseek-chat	No	66.40%
deepseek-reasoner	Yes	69.40%
gemini-2.0-flash	No	60.44%
gemini-2.5-flash	Yes	68.65%
gemini-2.5-flash*	Yes	87.60%

Table 2: Performance of API-based LLMs. All models were evaluated using a batched input of 50 questions, except where noted by an asterisk (*).

4 Conclusion

We presented a lightweight framework for automated Islamic inheritance reasoning (*’Ilm al-Mawārīth*), combining a specialized Arabic text encoder with an Attentive Relevance Scoring (ARS) mechanism for multiple-choice questions. Our local model, using MARBERT, achieved a test accuracy of 69.87%, which, while promising, is notably lower than the 87.60% reached by leading API-based LLMs like Gemini. This performance difference stems from our model's core design, which forgoes explicit, step-by-step symbolic reasoning in favor of efficient semantic matching. Despite this accuracy trade-off, our approach offers significant advantages in computational efficiency, on-device deployability, and data privacy, making it a viable solution for resource-constrained or offline applications.

These results highlight a critical trade-off between peak performance and practical usability. The current accuracy level positions our system as a valuable **assistive tool** for legal experts rather than a fully autonomous decision-maker, underscoring the necessity of human oversight in such high-stakes, rule-based domains. This demonstrates that lightweight, domain-adapted models remain highly relevant for specific use cases. Future work will directly aim to close the accuracy gap by integrating symbolic reasoning capabilities to handle the precise calculations inherent in inheritance law. We will also explore hybrid approaches that combine the efficiency of our lightweight model with the reasoning power of large models to achieve an optimal balance of performance and practicality.

Acknowledgement

The authors thank Mr. Arturo Argentieri from CNR-ISASI Italy for his technical contribution to the multi-GPU computing facilities. This research was partially funded by the Italian Ministry of University and Research (MUR) with the project "Future Artificial Intelligence Research —FAIR" Grant number PE0000013 CUP B53C22003630006.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *arXiv preprint arXiv:2102.10684*.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Khalid Arabi and Samira Hassan. 2023. Large language models in the arabic-speaking world: A case study on domain-specific challenges. *Journal of Natural Language Processing Arabia*, 5(2):45–61.

Salah Eddine Bekhouche, Azeddine Benlamoudi, Yazid Bounab, Fadi Dornaika, and Abdenour Hadid. 2025. Enhanced arabic text retrieval with attentive relevance scoring. *arXiv preprint arXiv:2507.23404*.

Abdessalam Bouchekif, Samer Rashwani, Emad Mohamed, Mutaz Al-Khatib, 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, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 Second Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Ibrahim A. El-Far. 2011. The islamic rules of inheritance. *Journal of Islamic Accounting and Business Research*, 2(1):7–23.

Dr. Abedeen Esmaeili. 2012. *The Islamic Law of Succession: A Practical Guide to the Laws of Faraid*. AS Noordeen.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

OpenAI. 2023. Gpt-4 technical report.

Arthur Phillips and Roland Knyvet Wilson. 1995. *A Treatise on the Muhammadan Law*. Kegan Paul International.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. *arXiv preprint arXiv:2007.13184*.

Gemini Team and 1 others. 2023. Gemini: A family of highly capable multimodal models.

A Detailed Training Objective

This section provides the detailed mathematical formulation of the three loss components used in our training objective. The total loss is defined as:

$$\mathcal{L}_{\text{total}} = 0.4 \cdot \mathcal{L}_{\text{cons}} + 0.4 \cdot \mathcal{L}_{\text{dyn}} + 0.2 \cdot \mathcal{L}_{\text{reg}} \tag{6}$$

A.1 Contrastive Loss ($\mathcal{L}_{\text{cons}}$)

We use an InfoNCE-based contrastive loss on the [CLS] token embeddings. This loss aims to pull the question embedding ($\mathbf{q}$) closer to the correct answer embedding ($\mathbf{c}^+$) and push it away from the five incorrect answer embeddings ($\mathbf{c}^-$).

$$\mathcal{L}_{\text{cons}} = -\tfrac{1}{B}\sum_{i=1}^{B}\log\left(\frac{e^{\text{sim}(\mathbf{q}_i,\mathbf{c}_i^+)}}{e^{\text{sim}(\mathbf{q}_i,\mathbf{c}_i^+)}+\sum_{j=1}^{5}e^{\text{sim}(\mathbf{q}_i,\mathbf{c}_{i,j}^-)}}\right) \tag{7}$$

where $\text{sim}(\mathbf{a},\mathbf{b}) = (\mathbf{a}^\top\mathbf{b})/\tau$. Here, $\mathbf{q}_i$ and $\mathbf{c}_i$ are the embeddings for the question and answers, and τ is a trainable temperature parameter.

A.2 Dynamic Relevance Loss ($\mathcal{L}_{\text{dyn}}$)

This loss directly supervises the final ARS scores (r) to ensure they are well-calibrated. It maximizes the score for the correct answer and minimizes the score for a randomly selected incorrect answer.

$$\mathcal{L}_{\text{dyn}} = -\frac{1}{B}\sum_{i=1}^{B}\left[\log(r_i^+ + \epsilon) + \log(1 - r_i^- + \epsilon)\right] \tag{8}$$

Here, r_i^+ and r_i^- are the sigmoid-activated ARS scores for the correct and a randomly chosen incorrect answer. The constant ϵ ensures numerical stability.

A.3 Relevance Score Logit Regularization ($\mathcal{L}_{\text{reg}}$)

To improve training stability, we apply a regularization loss on the raw, pre-sigmoid relevance scores (logits, s). This loss maximizes the variance of the logits within a batch, encouraging the model to use a wider dynamic range for its scores.

$$\mathcal{L}_{\text{reg}} = -(\text{Std}(s^+_{\text{batch}}) + \text{Std}(s^-_{\text{batch}})) \tag{9}$$

where s^+_{batch} and s^-_{batch} are the sets of logits for all correct and incorrect answers across the batch. We minimize the negative standard deviation, which is equivalent to maximizing the standard deviation.

CIS-RG at QIAS 2025 Shared Task: Chain of Thought Prompting and Finetuning for Enhancing Performance of LLMs on Islamic Legal Reasoning and its Mathematical Calculations

Osama Zaki[1,2]**, Asmaa Badawy**[2]**, Nada Elgewily**[2]**, Ahmed Sharaf**[2]

1. ROBOTAAR, UK
2. Sinai University, El-Arish, Egypt

Abstract

The work in this paper is related to the shared task QIAS2025. In this paper we continue in assessing large language models on Islamic legal reasoning. It is a challenging task because LLMs have not yet evolved (especially the open-source models) to solve complex reasoning problems or to perform mathematical calculations that require several steps. The LLMs need to comprehend the problem and to generate accurate and justified answers. In this paper we confirm the results and the analysis given in (Bouchekif, 2024; Bouchekif, 2025). However, we experiment further with Fine Tuning and Chain of Thought (CoT) to improve the performance of the reasoning process and therefor the results of the LLMs.

1 Introduction

This shared task assesses the ability of LLMs to accurately answer questions about ʿlm al-mawārīth (The science of Islamic Inheritance) in realistic scenarios. It is a major specialized topic in Islamic law. Islamic inheritance's rules are well defined, but it requires a complex reasoning mechanism and well-designed and systematic calculations procedures. There are mainly three computational stages, each includes zero, one or more than one step, that required to solve an Islamic inheritance case. First stage is to comprehend the inheritance scenario presented to the system, to identify eligible and the non-eligible heirs based on their relationship to the deceased person, bequests, the distribution of a defined amount of money, blocking or exclusion of some heirs, and to apply the basic fixed-share rules (farāʾiḍ). Second stage is to consider the cases where there are multiple heirs, multiple deceased individuals, residuary shares, and partial exclusion. Third stage is to consider the intricate fractional calculations, adjusting and redistribution, exaptational and nuanced cases, and juristic disputes. Although those stages seems like they can be carried out in sequence they are Intertwined and the system it has go back and forward over the rules. This makes the science of inheritance complex due to its diverse situations, the multiplicity of heirs, and the factors affecting the calculation of the estate, which requires a precise understanding of the texts of Islamic law and their correct application to prevent disputes and achieve justice in the distribution of rights.

2 System Design Issues

In (Bouchekif, 2024; Bouchekif, 2025) the performance of seven LLMs were assessed using a benchmark of 1,000 multiple-choice questions covering diverse inheritance scenarios, designed to test each model's ability to solve such problems. Gemini 2.5 and o3, demonstrated high performance, achieving accuracy above 90%. GPT4.5 achieved moderate results. Jais, Mistral, and LLaMA showed significantly lower

Proceedings of The Third Arabic Natural Language Processing Conference, pages 935–939
November 8-9, 2025 ©2025 Association for Computational Linguistics

accuracy reflecting their limitations in legal reasoning. There is a clear gap between models with reasoning abilities and those without. ALLaM, Fanar, LLaMA, and Mistral, consistently struggled with identifying complex familial relationships, evaluating diverse inheritance scenarios, and correctly executing corrective calculations.

As shown in the following section we assessed four models: Fanar, Llama, Gemini and Mistral. The models are further fine-tuned with a well-defined and large set of 1000 examples. We also recognized that the model architecture plays a major role in the result, i.e., being capable of performing reasoning or not. Models with reasoning capabilities consistently perform better. Having stated that, the reasoning capability is usually built outside the core of the model it is usually build at the application layer, i.e. the prompt, being the layer representing both the input and the output of the model. Many models nowadays claim that they have reasoning capability or at least able to respond correctly to simple reasoning task, but the challenge however among models present when dealing with complex reasoning problems.

LLMs evolved from just being a next-token prediction task dealing mainly with natural language (Zhao, 2023), to code generation (Gehring, 2024), and logical reasoning (Webb, 2023). Techniques such as chain of thoughts prompting techniques (Wei, 2022), tree of thought (Yao, 2024), trial-and-error search (Luo, 2024), Process Reward Models which facilitate reinforcement learning for LLMs (Sun, 2024). These emergent techniques are based on two main concepts in the traditional AI: "search" and "learning". A combination of scaling train-time compute and test-time-compute leads to better reasoning performance (OpenAI, 2024). To sum up, there is main four approaches for reasoning: 1) chain-of-thought (CoT) prompting which increases computational resources during inference to improve output quality. 2) Pure reinforcement learning (RL) 3) supervised finetuning (SFT) 4) combining both RL and SFT (Raschka, 2025).

In this paper, CoT prompting combined with finetuning is being the focus in our investigation. To do that, different finetuning datasets were prepared with different sizes (e.g., 100 and 200 questions) representing two different clusters. The first contains samples without any mathematical calculations, while the second contains samples that require mathematical calculations. As an example for the first cluster:

مات وترك: عم الأب لأب (4) وأخت لأب (5) و عم لأب (2) و
أم أم الأب وأم أم الأم كم النصيب الأصلي لـ عم لأب (2) من
التركة، وما الدليل على ذلك؟

The chain of thought (step-by-step) that should be followed is:

1. The type of actors in the question:
   وأم أم الأم ,أم أم الأب ,عم لأب ,أخت لأب ,عم الأب لأب
2. Those who deserve a fixed share:
   وأم أم الأم ,أخت لأب ,أخت لأب
3. Those who deserve a non-fixed share:
   عم لأب
4. Those who are blocked:
   عم الأب لأب
5. From the above,
   عم لأب هو من العصبات بالنفس، ويرث ما بقي من
   التركة بعد أصحاب الفروض
6. The number of actors in each type
   ما بقي من التركة يقسم على اثنين

To list the steps for each question (case) in this manner is unrealistic, but it is possible only for few shots. However, it still requires considerable efforts and skills to integrate CoT with the MCQs dataset.

Results of CoT promoting approach is still under investigation. In the following section we analysis the results of the traditional finetuning approach without the implementation of CoT.

3 Experimental Setup, Results and Analysis:

Four LLMs were fine-tuned: Fanar, Llama, Gemini and Mistral. The models were tested on the provided dataset by the shared task which contains unlabeled 1000 MCQs questions (answers is one of the letters: A, B, C, D, E or F, i.e. six choices).

The results are Gemini 2.5 and o3, demonstrated high performance, achieving accuracy above 90%. Fanar, Mistral, and LLaMA achieved moderate results 76%, 74%, 73% respectively reflecting their limitations in legal reasoning. This confirms to the findings in

Four examples form the test dataset are selected to demonstrate different scenarios, Table [1]. Our analysis shows that the level of the question being beginner or advanced has some impact on the model, but it is the wording of the question is the main reason that makes the model comprehend the questions. This means that classifying the questions to beginner and advanced are not very useful.

Question ID	Gold	Level	Fanar	Llama	Gemini	Mistral
8804_nl1d9s7s_4	E	Beginner	E	C	E	E
7398_nf5b3z3e_11	A	Advanced	D	C	A	C
8545_ni6z3l6j_12	D	Advanced	C	F	D	F
6839_np2c3f4i_4	B	Beginner	A	D	A	F

Table 1: Results from Fanar, Llama, Gemini and Mistral

But rather we can classify the questions into logical and mathematical problems. It is obvious that the commercial models are better pre-trained and hence fine-tunned. The generated justification and calculations of the commercial models were better represented and more detailed. The four selected example are (Fanar and Gemini only):

Question 8804_nl1d9s7s_4

مات وترك: عم الأب لأب (4) و أخت لأب (5) و عم لأب
(2) و أم أم الأب وأم أم الأم كم النصيب الأصلي لـ عم لأب
(2) من التركة، وما الدليل على ذلك؟
"نصيبه هو لا شيء، والدليل: لأنه عصبة","نصيبه هو
باقي التركة، والدليل: لأنه عصبة","نصيبه هو محجوب،
والدليل: لأنه عصبة","نصيبه هو كل التركة، والدليل: لأنه
عصبة","نصيبه هو باقي التركة، والدليل: لأنه
عصبة","نصيبه هو النصف، والدليل: لأنه عصبة"

Gemini

التعليل: عم لأب هو من العصبات بالنفس، ويرث ما بقي
من التركة بعد أصحاب الفروض (الأخت لأب والجدات).
وقد أجمع العلماء على أن العصبات يرثون ما فضل عن
نصيب أصحاب الفروض، عملاً بقول النبي صلى الله عليه
وسلم: "ألحقوا الفرائض بأهلها، فما بقي فلأولى رجل
ذكر". وهو هنا أقرب العصبات الموجودين في المسألة
(أقرب من عم الأب لأب) فيأخذ الباقي.

Fanar

التعليل: لأن العم لأب من العصبات، وعند وجود عصبة
من الذكور، يأخذون ما تبقى بعد الفرائض، ولا يوجد في
هذه المسألة ذكور عصبة غير العم لأب، لذا فهو يأخذ
الباقي من التركة. (ولكن يجب التنويه أن العم لأب ليس
أولى العصبات، بل يأتي بعد الأب والجد وأخو الأب وأم
الأب في الترتيب، ولكن في هذه المسألة، لا يوجد من هو
أولى منه من الذكور العصبة، لذا فهو يأخذ الباقي)

Three models were able to produce the correct answer. The Justifications given by Fanar and Gemini are also correct, but Gemini style is clearer. Although a fault was injected into the question (e.g., choices B and E), both exactly have the same wording, the three models have not detected that and give the answer as E.

Question 7398_nf5b3z3e_11

"مات وترك :أم وأخ لأم (3)و ابن ابن أخ لأب (2)و ابن ابن
(2)و ابن ابن أخ شقيق وأخ شقيق (4)، كم إجمالي عدد
الأسهم الذي تقسم عليه التركة قبل تصحيح المسألة؟",6
أسهم 0,سهم 7,أسهم 8,أسهم 4,أسهم 5,أسهم

Gemini

التعليل :الأم ترث السدس لوجود الفرع الوارث)ابن الابن(،
لقوله تعالى" :ولأبويه لكل واحد منهما السدس مما ترك إن
كان له ولد ."وابن الابن يحجب جميع الإخوة)الأخ الشقيق،
والأخ لأب، والأخ لأم (وأولادهم .وبناءً عليه، يكون أصل
المسألة من مقام نصيب الأم وهو 6أسهم .

Fanar

التعليل :يبدأ حساب الأسهم من أصل المسألة الذي يخرج منه
فرض الأم، وهو ثمانية أسهم، حيث تأخذ الأم الثمن)واحد (
والباقي للفرع الوارث تعصيباً

Only Gemini was able to generate the correct answer for this advanced question. The justification given by Gemini is correct and it is detailed, while justification from Fanar is incorrect. This shows that Fanar didn't comprehend the problem by saying:

وهو ثمانية أسهم، حيث تأخذ الأم الثمن (واحد) والباقي للفرع
الوارث تعصيباً ”

Question 8545_ni6z3l6j_12

مات وترك :أب الأب و أخ لأم (3)و أخ شقيق (2)، كم عدد
الأسهم بعد التصحيح التي يحصل عليها لكل أخ شقيق؟",4
أسهم,5 أسهم,0 سهم,1سهم,3أسهم,سهمان

Gemini

التعليل: في هذه المسألة، يحجب أب الأب (الجد الصحيح) الإخوة
لأم بالإجماع. أما بالنسبة للإخوة الأشقاء مع الجد، فإن مذهب
الجمهور يقضي بأن الجد يقاسم الإخوة أو يأخذ ثلث المال (أيهما
أفضل له). في هذه الحالة، لا يوجد أصحاب فروض آخرون،
فالجد يقاسم الإخوة الأشقاء (وعددهم 2).
إذا اعتبرنا أصل المسألة 3 سهام:
*   الجد يأخذ 1 سهم (3/1 من التركة).
*   الأخوان الشقيقان يأخذان 2 سهم (3/2 من التركة).
*   نصيب كل أخ شقيق = 2 سهم / 2 أخ = 1 سهم.

Fanar

التعليل :لا يرث الأخ لأم مع وجود الأخ الشقيق، فالأخ الشقيق
يحجب الأخ لأب .

Gemini was able to generate the correct answer for this advanced question. The justification and the explanation of mathematical steps given by Gemini are correct and clear which includes fractional allocation, while the justification from Fanar is incorrect which means Fanar has not understood this case (question).

Question 6839_np2c3f4i_4

مات وترك: عم الأب لأب و أم أم الأم و ابن ابن عم شقيق (3)و زوجـة كم النصيب الأصلي لـ ابن ابن عم شقيق (3) من التركة، وما الدليل على ذلك؟ ,"

نصيبه هو باقي التركة ، والدليل: ترتيب العصبات يكون أولا بالجهة ثم بالدرجة ثم بالقوة فمن حيث الجهة يقدم الأبناء الذكور على الأصول -الأب وأب الأب وإن علا - ، ومن حيث الدرجة يقدم من علا على من نزل فيقدم الإبن على ابن الإبن والأب على أب الأب ، ومن حيث القوة يقدم من كانت قرابته بأبوين على من كان بأب فقط فيقدم الأخ الشقيق على الذى لأب ويقدم ابن الأخ الشقيق على ابن الأخ لأب ، فلا يرث أبعد بتعصيب مع أقرب منه ، لقوله ﷺ (الحقوا الفرائض بأهلها فما بقى فهو لأولى رجل ذكر) وهنا أولى رجل ذكر هو ابن ابن عم شقيق",
" نصيبه هو باقي التركة، والدليل: ترتيب العصبات يكون..
" نصيبه هو باقي التركة، والدليل: ترتيب العصبات يكون...
" نصيبه هو باقي التركة، والدليل: ترتيب العصبات يكون...
" نصيبه هو باقي التركة، والدليل: ترتيب العصبات يكون...
" نصيبه هو باقي التركة، والدليل: ترتيب العصبات يكون

Gemini

تعليل: في هذه المسألة، لدينا أصحاب فروض وعصبات.
الزوجة: ترث الربع (4/1) لعدم وجود فرع وارث للميت، لقوله تعالى: "وَلَهُنَّ الرُّبُعُ مِمَّا تَرَكْتُمْ إِن لَّمْ يَكُن لَّكُمْ وَلَدٌ" (النساء: 12).
أم أم الأم: هي جدة صحيحة، وترث السدس (6/1) عند عدم الأم، وهو قول الجمهور.
ابن ابن عم شقيق (3) وعم الأب لأب: كلاهما من العصبات بالنفس. يتم تقديم العصبات بالترتيب: الجهة، ثم الدرجة ثم القوة.
كلاهما من جهة العمومة.
بين ابن ابن عم شقيق وعم الأب لأب، ابن ابن العم الشقيق أقرب في الدرجة والقوة. ابن ابن العم الشقيق هو من فروع أعمام الميت، بينما عم الأب لأب هو من أعمام أب الميت (أي أعمام الجد). الأقرب من جهة الميت يقدم، ففروع أعمام الميت مقدمون على أعمام الأب.
لذلك، يحجب عم الأب لأب بابن ابن عم شقيق.
يأخذ ابن ابن عم شقيق (3) باقي التركة بعد أصحاب الفروض.
الحساب:
أصل المسألة من 12 (المضاعف المشترك الأصغر للمقامات 4 و 6).
نصيب الزوجة: 12 ÷ 4 = 3 أسهم.
نصيب أم أم الأم: 12 ÷ 6 = 2 سهمين.
مجموع أسهم أصحاب الفروض: 3 + 2 = 5 أسهم.
الباقي للعصبة (ابن ابن عم شقيق (3)): 12 - 5 = 7 أسهم.
الدليل العام للعصبات هو قوله ﷺ: "ألحقوا الفرائض بأهلها فما بقي فهو لأولى رجل ذكر" (صحيح البخاري ومسلم).
وهنا أولى رجل ذكر هو ابن ابن عم شقيق.

Fanar

التعليل: ترتيب العصبات يكون أولا بالجهة، ثم بالدرجة ثم بالقوة، فمن حيث الجهة يقدم الأبناء الذكور على الأصول (الأب وأب الأب وإن علا)، ومن حيث الدرجة يقدم من علا على من نزل فيقدم الإبن على ابن الإبن والأب على أب الأب، ومن حيث القوة يقدم من كانت قرابته بأبوين على من كان بأب فقط فيقدم الأخ الشقيق على الذي لأب ويقدم ابن الأخ الشقيق على ابن الأخ لأب، فلا يرث أبعد بتعصيب مع أقرب منه، لقوله ﷺ (الحقوا الفرائض بأهلها فما بقي فلأولى رجل ذكر). وهنا أولى رجل ذكر هو ابن ابن عم شقيق، لذا يكون نصيبه باقي التركة .

This example demonstrates a basic question, but its prompt has some issues. The evidence referred to by each choice is identical for all of them. This

4 Conclusion and Future Work

Our experiments with finetuning have revealed some important points: 1) the classifications of questions into two or three categories (intermediate, advanced) have not shown noticeable impact on the results, 2) the uncleanses of the training or the validation dataset has some impact on the results, 3) LLMs with no reasoning capabilities (mostly the open source LLMs) struggle to solve complex reasoning problems, 4) MCQ is not the optimal option to train, validate especially when representing an ‘evidence’ with the ‘choice’, and this evidence is shared among other choices.

However, our initial experiments (ongoing work) with CoT have shown some promising results. We plan to combine CoT with RL. We intend classifying the questions (datasets) into two clusters: logical thinking problems and mathematical calculations problems.

We also believe a hybrid approach, agentic AI or neuro-symbolic systems, which can reason step by step, in algorithmic manner, that adhere exactly and precisely to legal rules and adapt to complex inheritance cases will enhance the performance.

Finally, when dealing with legal and/or religious domain such as the Islamic inheritance the LLMs responses should be verified by a legal lawyer or in a court.

References

Abdessalam Bouchekif, Samer Rashwani, Heba Sbahi, Shahd Gaben, Mutaz Al-Khatib, and Mohammed Ghaly. 2025. *Assessing Large Language Models on Islamic Legal Reasoning: Evidence from Inheritance Law Evaluation*, Proceedings of The Third Arabic Natural Language Processing Conference, Suzhou, China. Association for Computational Linguistics.

Abdessalam Bouchekif and Samer Rashwani and Emad Mohamed and Mutaz Al-Khatib and Heba Sbahi and Shahd Gaben and Wajdi Zaghouani and Aiman Erbad and Mohammed Ghaly. 2025. *QIAS 2025: Overview of the Shared Task on Islamic Inheritance Reasoning and Knowledge Assessment.* Proceedings of The Third Arabic Natural Language Processing Conference, ArabicNLP 2025, Suzhou, China, November 5--9, 2025. Association for Computational Linguistics.

Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Taco Cohen, and Gabriel Syn-naeve. 2024. Rlef: Grounding code llms in execution feedback with reinforcement learning. arXiv preprint arXiv:2410.02089.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, et al. 2024. Improve mathematical reasoning in language models by automated process supervision. arXiv preprint arXiv:2406.06592.

Sebastian Raschka. 2025. Understanding Reasoning LLMs. Ahead AI. https://magazine.sebastianraschka.com/p/understanding-reasoning-llms

Chuanneng Sun, Songjun Huang, and Dario Pompili. 2024. Retrieval-augmented hierarchical in-context reinforcement learning and hindsight modular reflections for task planning with llms.2024. OpenAI. 2024. Learning to reason with llms.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. Nature Human Behaviour, 7(9):1526–1541.

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.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems, 36.

Osama Zaki. 2024. Coupling Machine Learning with Ontology for Robotics Applications. Robotics. arXiv:2407.02500.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. arXiv preprint arXiv:2303.18223.

SEA-Team at QIAS 2025: Enhancing LLMs for Question Answering in Islamic Texts

Sanaa Alowaidi
University of Leeds
King Abdulaziz University
saalowaidi@kau.edu.sa

Eric Atwell
University of Leeds
e.s.atwell@leeds.ac.uk

Mohammed Ammar Alsalka
University of Leeds
em.a.alsalka@leeds.ac.uk

Abstract

This paper presents our participation in the QIAS 2025 shared tasks, namely Islamic Inheritance Reasoning and Islamic Knowledge Assessment sub-tasks. We propose an Islamic Retrieval-Augmented Generation (RAG) system that integrates multiple knowledge sources and semantic retrieval methods. Our evaluation compares multilingual general-purpose models and Arabic-centric models, using the accuracy metric. Results show that multilingual models consistently outperform Arabic-language models. The Mistral-large achieved the highest accuracy in Task 1 (72%) using basic RAG with our augmented knowledge resource, while GPT-4o with RAG and K2R retrieval achieved the best score in Task 2 (87.71%). These findings highlight the effectiveness of RAG in enhancing LLM performance for complex Islamic reasoning and knowledge assessment tasks.

1 Introduction

Large language models (LLMs) demonstrate strong capabilities in understanding, interpreting, and generating text that is close to human language. Several powerful multilingual general-purpose models have emerged, such as GPT-4o and Mistral-large. There are several Arabic-centric models developed recently, including Falcon (Almazrouei et al., 2023), ALLaM (Bari et al., 2024), Mistral SABA, and Fanar (Abbas et al., 2025), which are trained on specialized Arabic and Islamic knowledge resources. Arabic and religious texts present significant challenges for LLMs due to their linguistic complexity and the sensitive nature of Islamic teachings. Recently, Retrieval-Augmented Generation (RAG) has emerged as one of the most effective NLP techniques for question-answering. It enhances LLMs ability by retrieving relevant information from external knowledge sources and then using it to generate more accurate responses (Lewis et al., 2020; Oche et al., 2025). RAG is significant for domain-specific applications where accuracy and reliability are critical (Han et al., 2024).

Prior studies have applied RAG to various Islamic domains, including Quranic teachings, Turkish Islamic knowledge, and historical Islamic medical texts (Alnefaie et al., 2024; Alan et al., 2025; Sayeed et al., 2025). Moreover, promising research has focused recently on enhancing the retrieval stage of the RAG pipeline through query expansion and reformulation strategies in English (Yang et al., 2025; Wang et al., 2024).

To the best of our knowledge, RAG techniques have not yet been evaluated for Islamic inheritance reasoning or Islamic knowledge assessment. We address this gap by contributing to Task 1: Islamic Inheritance Reasoning and Task 2: Islamic Knowledge Assessment, as introduced in the QIAS 2025 shared task (Bouchekif et al., 2025a). Task 1 evaluates an LLMs ability to answer questions requiring precise reasoning and calculations based on Islamic jurisprudence. Task 2 assesses the accuracy of LLMs in answering general Islamic questions across multiple disciplines. Both tasks are challenging not only because they require advanced reasoning, but also because they include questions of varying difficulty levels that reflect the depth and complexity of Islamic knowledge. In this paper, we investigate strategies to enhance LLMs for Islamic QA, addressing the following research questions: how does the combination of few-shot prompting and RAG techniques affect the LLMs? How does the type of LLM, general multilingual or Arabic-centric, affect the accuracy of RAG? Does the size of the knowledge resource affect the performance of the RAG system? What is the effect of applying semantic retrievals through query expansions and reformations on LLMs?

Proceedings of The Third Arabic Natural Language Processing Conference, pages 940–946
November 8-9, 2025 ©2025 Association for Computational Linguistics

The paper is organized as follows: In Section 2, related works are reviewed. Section 3 details the datasets used. Section 4 describes the proposed system structure, while Section 5 describes the implementation setup. In Section 6, the results are presented and discussed. Finally, the paper concludes with a summary and suggestions for future work.

2 Related Works

Several studies have examined the application of LLMs to Islamic knowledge. Alnefaie et al. (2023) used GPT for question answering on a Quran dataset. Bouchekif et al. (2025b) evaluated several multilingual LLMs and Arabic LLMs with zero-shot prompting on an inheritance dataset. These works highlighted key limitations of LLMs, including hallucination and misinterpretation. More recent research has explored using RAG techniques to improve LLM performance. Alnefaie et al. (2024) applied RAG to the GPT-4 model in the Quranic Question Answer dataset. Alan et al. (2025) introduced the MufassirQA system, which enhances the ChatGPT-3.5 Turbo model with RAG by using religious knowledge resources in the Turkish language. Furthermore, Sayeed et al. (2025) investigated the use of RAG with LLaMA-3, Mistral-7B, and Qwen-2 to answer medical questions based on an old Islamic medical text. These studies found that RAG consistently outperforms baseline LLMs and emphasized that its performance is highly dependent on the quality of the retrieval and knowledge resources. However, the performance of RAG in Islamic domain-specific knowledge remains largely underexplored. Furthermore, promising research has recently focused on improving the retrieval stage of the RAG pipeline through query expansion and reformulation strategies in English (Yang et al., 2025; Wang et al., 2024; Li et al., 2024). In this study, we extensively explore the RAG in both Arabic and multilingual LLMs. In addition, study the effect of different retrieval strategies that incorporate query expansion and reformulation methods.

3 Datasets

In this paper, we use the four officially published datasets [1] corresponding to the two subtasks of the QIAS 2025 shared task.

3.1 Task 1: Islamic Inheritance Reasoning (Ilm al-Mawrth)

The Islamic Inheritance dataset comprises 22,000 multiple-choice questions (MCQs). Each question includes six answer choices with only one correct label (Bouchekif et al., 2025b). The questions are classified into two levels of difficulty: beginner and advanced. The dataset was divided into 20,000 for the training set, 1,000 for the validation set, and 1,000 for the test set. In addition, the fatwa dataset is used as a supplementary knowledge resource. It consists of 3165 fatwas from Islamic websites covering general legal, ethical, and social topics.

3.2 Task 2: General Islamic Knowledge

The first dataset consists of 1700 question pairs in MCQ format covering Hadith criticism, Quranic sciences, legal theory, and prophetic biography. Each question has four answer choices, with one correct answer. The data distribution is 700 question pairs for the validation set and 1,000 for the test set. The questions are categorized into three complexity levels: beginner, intermediate, and advanced. Moreover, a supplementary Islamic corpus was used as an external knowledge resource for the RAG system. It comprises unsupervised data of relevant Islamic texts. The corpus includes approximately 50 Islamic books in MS Word format, all of which are directly related to the evaluation dataset topics.

4 System Overview

The proposed system adopts the RAG architecture (Lewis et al., 2020; Wang et al., 2024; Oche et al., 2025) and consists of three main phases [2]: Knowledge Resource Preparation, Retrieval, and Answer Generation, as illustrated in Appendix A. The Knowledge Resource Preparation phase is conducted offline, where documents are preprocessed and converted into vector representations. This phase includes four modules: loading, chunking, embedding, and indexing. First, the input documents are loaded and preprocessed to produce normalized, cleaned text. Next, the chunking module divides the documents into smaller units. This step is essential for improving retrieval effectiveness, enabling embedding storage, and addressing the

[1] https://gitlab.com/islamgpt1/qias_shared_task_2025

[2] in The code is available in our repository:https://github.com/S-Alowaidi/SEA-RAG_Enhancing-LLMs

context-length limitations of LLMs. In our experiments, we used token-aware recursive chunking to segment documents into semantically coherent units, ensuring token compatibility with the embedding model's tokenization. Following the recommendations in (Wang et al., 2024), we set the splitter to 500 tokens per chunk with a 50-token overlap. The embedding module then transforms each chunk into a high-dimensional dense vector using OpenAI Embeddings, enabling efficient semantic similarity searches. Finally, the indexing module stores these embeddings in a FAISS (Facebook AI Similarity Search) vector database. The Retrieval and Answer Generation phases are executed in real time. At query time, the question is embedded, and the retrieval module searches for the most relevant chunks in the vector index. We propose three semantic retrieval methods: basic similarity search and two semantically enhanced strategies. For the enhanced methods, keywords are extracted offline using GPT-4o. Candidate keywords are filtered to remove noise by eliminating stop words, short terms, and duplicates, as well as semantically irrelevant items using cosine similarity (threshold = 0.3) against the original question. Basic Similarity Search: retrieves chunks in a single pass using the original query. Keyword-Augmented Two-Stage Retrieval (K2R): performs parallel retrieval using the original query and semantically filtered keywords, then merges and deduplicates the retrieved chunks. Multi-Query Reformulation with Keywords (MQR-K): reformulates each keyword with the query into a complete sub-question in Arabic, retrieves semantically similar chunks in parallel, and merges the results for diversification. In the Answer Generation phase, the retrieved context is combined with the question and its answer choices in a structured prompt, which is then sent to the LLMs. The output undergoes a post-generation validation step to ensure compliance with the single-letter MCQ answer format.

5 Experimental Setup

All experiments were run on Google Colab and used the LangChain framework. Embeddings are generated using text-embedding-3-large (OpenAI) and stored in a FAISS flat index with cosine similarity. The retrieval module is configured to return the top-$k = 4$ chunks per query. To address the third research question, in Task 1, we evaluate two datasets. The first is the fatwa dataset as a baseline knowledge resource for RAG. In addition, we proposed an expanded dataset that combines the fatwa dataset with MCQ training data by including question and correct-answer pairs as additional contextual knowledge. In the generation phase, we evaluate several LLMs in three-shot prompting, including GPT-4o [3], Mistral-large-latest [4], Fanar Islamic-RAG [5], Allam-7B [6], and Mistral-SABA-24B [7].

Model	3-Shot	Fatwa	Expand-K	K2R
Fanar	53.3	57.8	62.8	59.5
Mistral-S	43.8	50.1	55.6	53.2
Mistral-L	61.5	66.0	72.0	70.0
ALLaM	47.8	50.6	53.2	43.5
GPT-4o	57.5	58.9	61.6	63.4

Table 1: Task 1 results comparing Few-Shot Prompting, Basic RAG (Fatwa only), Expanded Knowledge (Fatwa + Train), and RAG with K2R.

Model	Few-Shot	Basic	K2R	MQR
Fanar	29.43	57.86	54.14	55.71
Mistral-S	66.07	75.86	76.29	72.43
Mistral-L	78.29	83.86	77.57	76.00
ALLaM	71.29	77.29	79.43	77.43
GPT-4o	83.71	83.86	87.71	85.57

Table 2: Task 2 results comparing Few-Shot prompting, Basic RAG, K2R retrieval RAG, and MQR-K retrieval RAG.

6 Results and Discussion

Task 1: Table 1 shows the results on the development set. The Mistral-large model achieved the highest accuracy 72.0% when using RAG with the expanded fatwa dataset. Comparing the fatwa-only dataset to the augmented version, the Arabic-centric models Mistral-SABA-24B and Fanar Islamic-RAG benefited the most, with gains of 6.3 and 4.5 points, respectively. ALLaM-7B showed the least improvement. On the other hand, Mistral-large had the second-highest gain 6.0 points, while GPT-4o's improvement was relatively small at 2.7 points. This highlights how the reasoning ability of multilingual models can be greatly enhanced by adding domain-specific knowledge. **Test set:** For the test set, we selected

[3] via OpenAI API:`https://platform.openai.com/`

[4] via Mistral API:`https://mistral.ai/`

[5] via Fanar API:`https://fanar.qa/`

[6] via: `https://huggingface.co/transformers`

[7] via Groq API

Model	T1Ed	T1K2R	T2R	T2K2R
Mistral-L	61.1	63.0	87.7	89.1
GPT-4o	59.9	57.1	87.8	89.0

Table 3: Results on test set for Task1: T1Ed refers to Expanded Knowledge,T1K2R refers to K2R retrieval RAG. Task2:T2R refers to Basic RAG, T2K2R refers to K2R retrieval RAG

the best-performing approaches based on the results from the development set. As presented in Table 3, Mistral-large achieved the highest accuracy using the K2R method, reaching 63%. Unlike the development set, its performance improved in the test set. In contrast, GPT's performance with the K2R method declined slightly in the test data.

Task2:

Table 2 shows the accuracy results for different retrieval methods in answering general Islamic questions. GPT-4o achieved the highest accuracy of 87.71% using the K2R retrieval method, outperforming its baseline RAG. This is expected since GPT-4o has been trained on a large amount of data, including Islamic knowledge. Mistral-large achieved the second-highest accuracy in the baseline RAG 83.86%. However, its performance dropped slightly with the K2R and MQR-K retrieval methods (77.57% and 76.00%, respectively. The performance of Arabic-centric models varied across retrieval methods. ALLaM-7B and Mistral-SABA performed best with K2R, while Fanar achieved its best results, 55.71%, with MQR-K.

Test set: For the test set, we chose two strategies based on their performance during the development phase. The table 3 indicates that Mistral-large and GPT-4o achieved very similar results, both reaching approximately 89% with the K2R method. Therefore, the expanded query could be a promising approach to enhancing RAG.

Overall Analysis Based on the results, it is clear that RAG performance is heavily dependent on the quality of the retrieved contexts. Enhancing retrieval with the K2K approach outperformed basic RAG retrieval for all models. However, the performance continued to fluctuate compared to the knowledge-enrichment approach and depended on the nature of the task and the model used. For example, in task 1, the nature of inheritance law texts often shares similar keywords (e.g., wife, paternal mother, heirs). Hence, refining the query by broadening keywords could produce a wider context that distracts the LLMs. In the case of multiple-query reformulation (MQR), we observed a general drop in performance for most models, except Fanar, which showed a slight improvement. This may be due to the static reformulation method used, which can cause loss of semantic meaning and introduce noise. The results show that, in general, Arabic-centric models benefit from higher recall when broadening the context by expanding queries with keywords. In contrast, stronger models perform better with fewer but more relevant contexts.

7 Conclusion

This work presents our contributions to the QIAS 2025 shared task, focusing on Task 1: Islamic Inheritance Reasoning and Task 2: Islamic Knowledge Assessment. We propose an Islamic RAG system that leverages multiple knowledge sources and retrieval methods, utilizing more than five different LLMs. Our experimental results show that multilingual general-purpose models outperform Arabic-language models in both tasks. For Task 1, Mistral-large achieved the best performance (72%), while for Task 2, GPT-4o delivered the strongest results in general Islamic knowledge reaching (87.71%). Among the Arabic models, Fanar performed best in Task 1 by 62.8%, and ALLaM-7B led in Task 2 by 79.43%. We also observed that expanding the knowledge sources in Task 1 improved the performance of all models, with the most notable gains for Arabic models such as Fanar and ALLaM-7B.

Regarding the use of RAG with a semantic retrieval strategy, results indicate that semantic retrieval RAG generally outperformed three-shot prompting across all models and both tasks. However, its advantage over basic RAG varied according to the nature of the task data and the model.

Future research should explore alternative query expansions and reformulation approaches, such as using LLMs to generate more semantically relevant queries dynamically. In addition, investigating other RAG enhancement techniques, including re-ranking and document summarization, may yield further improvements. Finally, we emphasize the importance of developing high-quality Islamic knowledge sources to improve model relearning effectively.

Acknowledgments

The authors would like to thank the competition organizers. We also thank KAU for its support.

References

Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur A. Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed K. Elmagarmid, Mohamed Y. Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, and 22 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydın. 2025. Improving llm reliability with rag in religious question-answering: Mufassirqas. *Turkish Journal of Engineering*, 9(3):544–559.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Is GPT-4 a good islamic expert for answering Quran questions? In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 124–133, Taipei City, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Sarah Alnefaie, Eric Atwell, and Mohammed Ammar Alsalka. 2024. Using the retrieval-augmented generation technique to improve the performance of gpt-4 in answering quran questions. In *2024 6th International Conference on Natural Language Processing (ICNLP)*, pages 377–381.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan AlRashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, and 6 others. 2024. Allam: Large language models for arabic and english. *Preprint*, arXiv:2407.15390.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Binglan Han, Teo Susnjak, and Anuradha Mathrani. 2024. Automating systematic literature reviews with retrieval-augmented generation: a comprehensive overview. *Applied Sciences*, 14(19):9103.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Zhicong Li, Jiahao Wang, Zhishu Jiang, Hangyu Mao, Zhongxia Chen, Jiazhen Du, Yuanxing Zhang, Fuzheng Zhang, Di Zhang, and Yong Liu. 2024. Dmqr-rag: Diverse multi-query rewriting for rag. *arXiv preprint arXiv:2411.13154*.

Agada Joseph Oche, Ademola Glory Folashade, Tirthankar Ghosal, and Arpan Biswas. 2025. A systematic review of key retrieval-augmented generation (rag) systems: Progress, gaps, and future directions. *arXiv preprint arXiv:2507.18910*.

Mohammad Amaan Sayeed, Mohammed Talha Alam, Raza Imam, Shahab Saquib Sohail, and Amir Hussain. 2025. From rag to agentic: Validating islamic-medicine responses with llm agents. *arXiv preprint arXiv:2506.15911*.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. 2024. Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17716–17736, Miami, Florida, USA. Association for Computational Linguistics.

Qimin Yang, Huan Zuo, Runqi Su, Hanyinghong Su, Tangyi Zeng, Huimei Zhou, Rongsheng Wang, Jiexin Chen, Yijun Lin, Zhiyi Chen, and Tao Tan. 2025. Dual retrieving and ranking medical large language model with retrieval augmented generation. *Scientific Reports*, 15(1):18062.

A System Architecture

A comprehensive description of the proposed RAG system is illustrated in Figure 1.

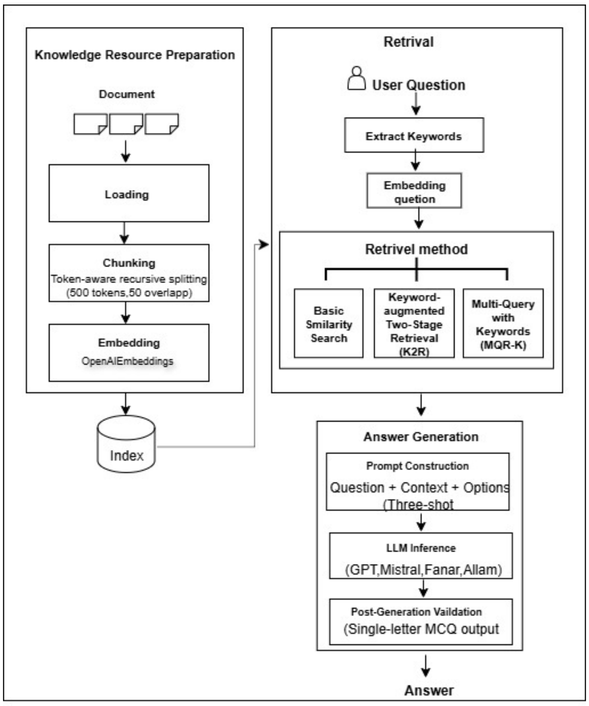
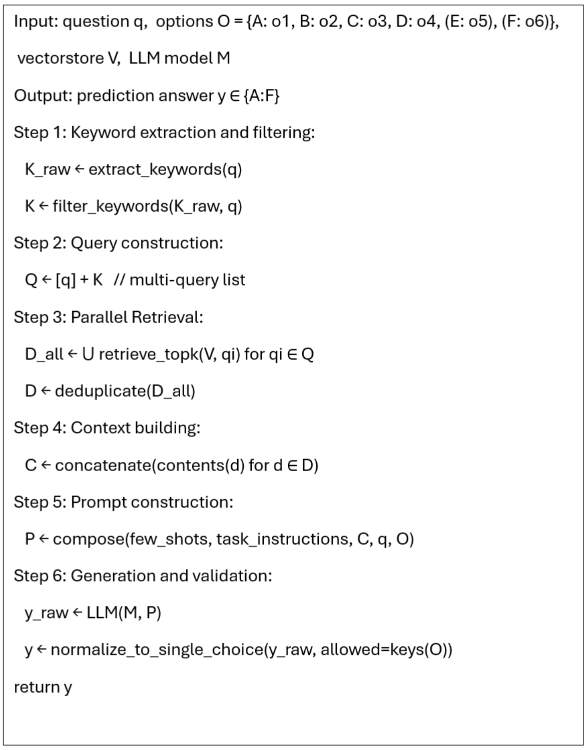

Knowledge Resource Preparation
Document
Loading
Chunking
Token-aware recursive splitting
(500 tokens,50 overlapp)
Embedding
OpenAIEmbeddings
Index
Retrival
User Question
Extract Keywords
Embedding quetion
Retrivel method
Basic Similarity Search
Keyword-augmented Two-Stage Retrieval (K2R)
Multi-Query with Keywords (MQR-K)
Answer Generation
Prompt Construction
Question + Context + Options (Three-shot
LLM Inference
(GPT,Mistral,Fanar,Allam)
Post-Generation Vaildation
(Single-letter MCQ output
Answer

Figure 1: The proposed RAG system architecture



```
Input: question q, options O = {A: o1, B: o2, C: o3, D: o4, (E: o5), (F: o6)},
vectorstore V, LLM model M
Output: prediction answer y ∈ {A:F}
Step 1: Keyword extraction and filtering:
  K_raw ← extract_keywords(q)
  K ← filter_keywords(K_raw, q)
Step 2: Query construction:
  Q ← [q] + K  // multi-query list
Step 3: Parallel Retrieval:
  D_all ← U retrieve_topk(V, qi) for qi ∈ Q
  D ← deduplicate(D_all)
Step 4: Context building:
  C ← concatenate(contents(d) for d ∈ D)
Step 5: Prompt construction:
  P ← compose(few_shots, task_instructions, C, q, O)
Step 6: Generation and validation:
  y_raw ← LLM(M, P)
  y ← normalize_to_single_choice(y_raw, allowed=keys(O))
return y
```

Figure 2: RAG based on the Keyword-Augmented Two-Stage Retrieval (K2R) approach

B Keyword-Augmented Two-Stage Retrieval (K2R)

The K2R approach retrieves documents using a multi-query parallel FAISS search. Figure 2 describes the general steps for RAG based on the K2R method.

C Multi-Query Reformulation with Keywords (MQR-K)

This approach is based on reformulating the question using a fixed Arabic template to generate one query per keyword. Figure 3 explains the general steps for RAG based on the MQR-K method.

Model	Beg.	Int.	Adv.
Mistral-L | T2R	90.57	85.33	76.67
GPT-4o | T2R	90.29	90.00	74.00
Mistral-L | T2K2R	92.29	85.33	78.00
GPT-4o | T2K2R	92.57	87.33	74.00
Mistral-L | T1Ed	75.20	–	47.00
GPT-4o | T1Ed	72.60	–	47.20
Mistral-L | T1K2R	78.80	–	47.20
GPT-4o | T1K2R	77.40	–	36.80

Table 4: Accuracy (%) across difficulty levels Beginner (Beg.), Intermediate (Int.), Advanced (Adv.). A dash (–) indicates an unavailable level. For Task 1: T1Ed refers to Expanded Knowledge, T1K2R refers to K2R retrieval RAG. Task 2:T2R refers to Basic RAG, T2K2R refers to K2R retrieval RAG

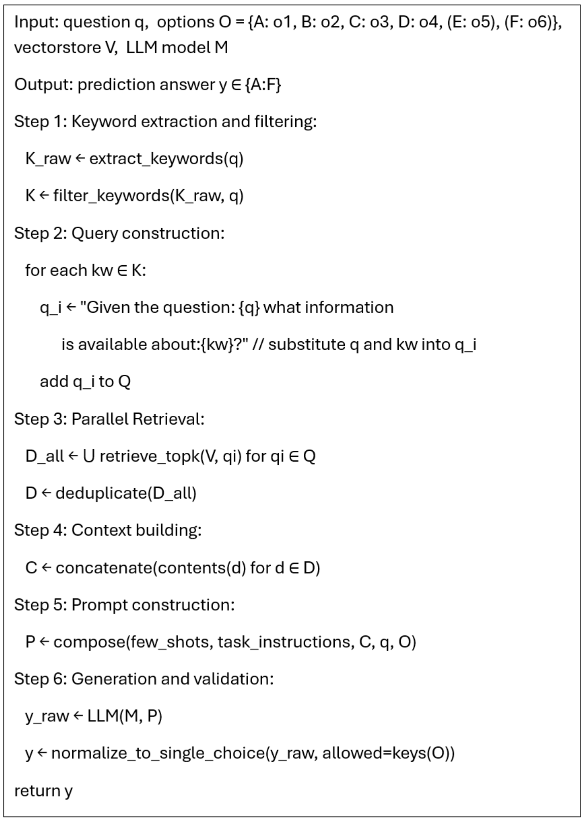
D Prompt

Figure 4 demonstrates the prompt used in the experiments. The few-shot examples refer to three examples specifically for the target task taken from the training data. The context refers to the documents retrieved using one of the retrieval methods, baseline semantic similarity, K2R, or MQR.

E In-depth Analysis

Table 4 presents the performance of various models on Task 1 and Task 2 across three difficulty levels: beginner, intermediate, and advanced. The results indicate that all models generally achieved high accuracy on beginner-level questions for both tasks.

In Task 1, the Mistral-large model answered approximately 75.20% of beginner questions, while the GPT-4o model answered about 72.60% when applying the RAG with the expanding knowledge approach. However, for advanced questions, the accuracy of most methods in answering these questions reaches only about 47%, indicating weaker performance on questions that require complex inheritance reasoning compared with simpler ones.



```
Input: question q,  options O = {A: o1, B: o2, C: o3, D: o4, (E: o5), (F: o6)},
vectorstore V,  LLM model M
Output: prediction answer y ∈ {A:F}
Step 1: Keyword extraction and filtering:
  K_raw ← extract_keywords(q)
  K ← filter_keywords(K_raw, q)
Step 2: Query construction:
  for each kw ∈ K:
     q_i ← "Given the question: {q} what information
          is available about:{kw}?" // substitute q and kw into q_i
     add q_i to Q
Step 3: Parallel Retrieval:
  D_all ← U retrieve_topk(V, qi) for qi ∈ Q
  D ← deduplicate(D_all)
Step 4: Context building:
  C ← concatenate(contents(d) for d ∈ D)
Step 5: Prompt construction:
  P ← compose(few_shots, task_instructions, C, q, O)
Step 6: Generation and validation:
  y_raw ← LLM(M, P)
  y ← normalize_to_single_choice(y_raw, allowed=keys(O))
return y
```

Figure 3: RAG based on the Multi-Query Reformulation with Keywords (MQR-K) approach

Additionally, the K2R retrieval RAG approach made substantial improvements on beginner-level questions. In contrast, for advanced questions, while the Mistral-large model maintained its accuracy in the K2R approach, the performance of the GPT-4o model decreased when queries were expanded with keywords.

For task 2, which focused on general Islamic knowledge, most approaches demonstrated exceptional performance, achieving accuracy rates of 92.57%, 90%, and 78% at the beginner, intermediate, and advanced levels, respectively. It is clear from the results that the K2R retrieval method achieved a notable improvement at the beginner and advanced levels across models. Moreover, the results show that while Mistral-large and GPT-4o both performed similarly overall, the Mistral-large model often slightly outperformed the GPT-4o model on advanced questions.

```
prompt = f"""
{few_shot_examples}
Context:
{context}
You are a specialist in Islamic sciences.
Your task is to answer multiple-choice
questions by selecting the correct option.
Question:
{question}
{options_text}
Please respond using only one English letter from:
{valid_letters}
Do not write any explanation or additional text.
""".strip()
```

Figure 4: The prompt used for the RAG system

MorAI at QIAS 2025: Collaborative LLM via Voting and Retrieval-Augmented Generation for Solving Complex Inheritance Problems

Jihad R'baiti[1]*, Chouaib EL Hachimi[2], Youssef Hmamouche[1], Amal El Fallah Seghrouchni[1,3]

[1] International Artificial Intelligence Center of Morocco (Ai movement), Mohammed VI Polytechnic University (UM6P), Rabat, Morocco.
[2] Research Centre for Artificial Intelligence in Geomatics (RCAIG), Department of Land Surveying and Geo-Informatics (LSGI), The Hong Kong Polytechnic University (PolyU), Kowloon, Hong Kong SAR.
[3] LIP6 - UMR 7606 CNRS, Sorbonne University, Paris, France.

Abstract

Collaborative approaches have proven effective in addressing complex problems, from human and socio-economic challenges to multi-agent systems. These methods rely on the principle that combining perspectives enhances problem-solving. In this paper, we propose a collaborative large language models (LLM) framework to solve Islamic inheritance problems, which demand precise mathematical reasoning and strict adherence to legal rules for fair distribution among heirs. The system implements a collaborative voting mechanism involving multiple LLMs, namely ALLaM-7B-Instruct-preview, Deepseek-reasoner, and Gemini-2.5-Flash. Each independently answered multiple-choice inheritance questions. The final answer is determined by majority vote. To improve accuracy and domain grounding, we integrate Retrieval-Augmented Generation (RAG). A curated database of solved inheritance cases in JSON format is indexed using TF-IDF. For each query, the most similar cases are retrieved and appended as contextual information to the prompt before being submitted to the LLMs. Experimental results demonstrate that this collaborative RAG-enhanced framework outperforms individual LLMs. The ensemble achieved 88% accuracy, surpassing the best-performing single models: the finetuned ALLaM-7B-Instruct-preview (79.50%), Deepeek-reasoner (71.80%), and Gemini-2.5-Flash (83.50%).

1 Introduction

LLMs have rapidly gained a prominent role in Natural Language Processing (NLP), transforming how machines understand and generate human-like language. From general-purpose systems like GPT (Achiam et al., 2023) and Gemini (Comanici et al., 2025) to Arabic-focused models such as Fanar (Team et al., 2025) and ALLAM (Bari et al., 2024b), these models have demonstrated remarkable proficiency across a wide range of tasks (Demidova et al., 2024; Singhal et al., 2025; Miah et al., 2024). They are now being used to solve open-domain problems, answer complex questions, and support more specialized areas that require structured knowledge and contextual understanding—such as legal, medical, or religious domains. One such domain is Islamic inheritance, which is based on detailed rules for distributing assets among heirs. Answering questions in this area requires an understanding of Islamic sources and the ability to perform precise calculations involving predetermined shares assigned to each heir. In many cases, small changes in family composition can lead to entirely different outcomes. This makes it a valuable challenge for testing how well LLMs can handle structured reasoning, numerical logic, and Arabic-language understanding in a religious context. To support research in this area, the Question-and-Answer in Islamic Studies Assessment Shared Task (QIAS 2025) (Bouchekif et al., 2025a) was introduced as a benchmark for evaluating the reasoning capabilities of LLMs in the domain of Islamic knowledge. Our focus is on Subtask 1: Islamic Inheritance Reasoning, which presents multiple-choice questions (MCQs) in Arabic across three levels of difficulty: beginner, intermediate, and advanced. The questions are designed to evaluate an LLM's understanding of Islamic inheritance principles and its ability to apply them accurately to solve practical cases. In our final submission, we developed an ensemble-based system that combines RAG with multiple pretrained LLMs to tackle the task's multiple-choice reasoning challenges. We used five models—ALLAM, Fanar, Qween, Gemini, and Deepseek—and applied prompting with RAG during inference. Each model independently predicts an answer (from A to F), and a majority voting strategy is used to select the final response. This setup leverages the con-

[*] Corresponding author: jihad.rbaiti@um6p.ma

Proceedings of The Third Arabic Natural Language Processing Conference, pages 947–952
November 8-9, 2025 ©2025 Association for Computational Linguistics

textual strength of RAG and the diverse reasoning capabilities of the models, enabling more robust performance across different question types and difficulty levels. Our system ranked 5th in Subtask 1 of the QIAS 2025 shared task. The full implementation is available at [1]. This paper is organized as follows:
Section 2 provides background relevant to our work. Section 3 describes the dataset, Section 4 presents our method. Section 5 detail the experimental setup used for evaluation. Section 6 presents the result obtained, and Section 7 concludes our work.

2 Background

2.1 Related Work

Recent work has explored the use of LLMs in religious domains, particularly in the Islamic context. (Mohammed et al., 2025) have proposed an advanced RAG approach using a re-ranker. This approach has proven effective, providing increased response stability, eliminating hallucinations, and obtaining a more accurate answer compared to both base LLM and LLM with RAG methodologies. Similarly, Alan et al. (2025) presented 'MufassirQAS', a system proposed to improve LLM transparency and accuracy using RAG. This system presented relevant sections from the retrieved database alongside the LLM's answers. While Akkila and Naser (2016) developed an expert system designed to simplify and automate the calculation of Islamic inheritance shares based on Sharia law, replacing the traditional method of calculation, aiming to reduce human error and disputes among heirs. In (Bouchekif et al., 2025b), the authors assess LLMs' reasoning capabilities in Islamic inheritance law. The results reveal o3 and Gemini 2.5 as the more accurate models, surpassing ALLaM, Fanar, and Mistral in terms of accuracy.

2.2 Islamic inheritance law

Islamic inheritance law (Ilm al-Mawārīth) is a rule-based, mathematical framework derived primarily from Surah An-Nisā' in the Qur'an, which specifies fixed fractional shares for eligible heirs. The framework considers three key factors when distributing shares:

- Degree of kinship: Closer relatives inherit larger shares regardless of gender;
- Generational position: Younger generations preparing for life's responsibilities receive larger shares than older generations relinquishing them;
- Financial responsibility: The only case where gender-based difference applies, when sons inherit twice the share of daughters due to lifelong financial support to their wives and families, while daughters retain their inheritance solely for themselves without any spending obligation.

Verses 4:11–12 and 4:176 outline concise, efficient, and generalizable distribution principles. Children inherit with males receiving the share of two females; parents receive one-sixth each if the deceased has children, with the mother's share adjusted in the presence of siblings; spouses inherit fixed portions depending on the presence of children; and siblings inherit in kalālah, which is the case when there are no ascendants or descendants, with males receiving twice the share of females. These rules are applied only after paying debts and executing valid bequests, which cannot exceed one-third of the estate.

3 Data

The approach uses two data sources (Bouchekif et al., 2025a): a structured dataset and a semi-structured dataset. The first one consists of MCQs. It was constructed by converting religio-ethical advice 'fatwas' collected from IslamWeb [2] into a structured format. The preprocessing included reviewing the MCQ by four experts in Islamic studies, rephrasing ambiguous questions, and eliminating semantic and numerical redundancies. Each MCQ has six answer options (A to F), with only one correct. The dataset design uses an increasing complexity of three levels of difficulty—beginner, intermediate, and advanced—to assess LLMs' capability across different levels of expertise. The dataset contains approximately 20000 MCQs for training, 1000 for validation, and 1000 for testing. On the other hand, the semi-structured dataset consists of 4 JSON files containing all necessary information about the problem statements 'fatwas', including ID, URL, category, Gregorian and Hijri dates, question, and answer. It was sourced from IslamWeb and provides about 3065 resolved inheritance problem statements. This dataset was used as

[1] https://shorturl.at/Fev1p

[2] https://www.islamweb.net/

an external source of knowledge to retrieve pertinent similar inheritance cases in our proposed RAG system.

4 System Overview

The proposed framework integrates a Multi-LLM Voting Framework with a Retrieval Module to address Arabic Islamic inheritance MCQs (Figure 1). First, a curated JSON knowledge base of solved inheritance cases 'Fatwas' is built and indexed using TF-IDF. After storing the index, each problem statement is vectorized as a query for the RAG system and projected in the index vector space model to be compared with the corpus's documents using cosine similarity, retrieving the top-k most similar cases. These retrieved cases are appended to the original problem statement prompt (Figure 2). This prompt augmentation enriches the context during the inference phase. The augmented prompt is then passed to multiple LLMs, which are the fine-tuned ALLaM-7B, Gemini-2.5-Flash, and DeepSeek-R1. The models independently generate their answers. Finally, the outputs are processed through a voting mechanism that selects the majority answer as the final prediction. In cases where no majority is reached, priority is given to Gemini, as it demonstrated the most reliable performance during experiments.

5 Experimental Setup

The goal of the shared task was to evaluate the ability of various LLMs to solve Islamic inheritance reasoning problems, which involve applying strict legal mathematical rules. Multiple submission strategies (Table 1) were developed and evaluated. The first approach involved supervised fine-tuning of the 7 billion version of the ALLaM model, which was selected for its reported strong performance in Arabic (Bari et al., 2024a). The training data consisted of inheritance cases formatted in a question-answering style described in Section 3. The model was trained using standard cross-entropy loss to directly predict the correct inheritance distribution, using the listed hyperparameters (Figure 2). The second approach uses zero-shot learning, where a prompt-based inference was applied without additional fine-tuning. Here, multiple pretrained LLMs were tested, including variants of ALLaM, Fanar, Gemini, Qween, and DeepSeek. Each model was prompted using a fixed template describing the inheritance case and requesting a response in a structured format. Next, a RAG setup was implemented to provide contextual fatwa-based information. Relevant fatwas were retrieved for each case and appended to the original prompt. Two retrieval methods were compared: one using a Neural Embedding for semantic search, and another using TF-IDF for keyword-based retrieval. The same prompting strategy was applied in both cases. Finally, based on model performance, the study chose 3 models to construct the pool of voting to implement the proposed majority voting collaborative strategy.

During implementation of RAG, we experimented with different values of top-k voting using k $\in \{3, 5, 7\}$, to evaluate the effect of the number of retrieved documents (k) and retrieval method (TF-IDF vs. Neural Embeddings). The combination of k=3 using TF-IDF was selected for a balance between computation, prompt input length, and model performance. For the inference, we applied parameter-efficient tuning via LoRA with rank set to 8, alpha set to 16, and dropout set to 0.05, using a maximum input length of 3000 tokens. Decoding was performed with beam search (num_beams=5) combined with sampling (temperature=0.6, top_p=0.9), and the maximum number of generated tokens was limited to 20.

This study uses the Mohammed VI Polytechnic University's high-performance computing (HPC) called 'TOUBKAL', which provides a cluster of various types of computational nodes equipped with NVIDIA A100-SXM4-80GB GPUs. To access the HPC, the MobaXterm software is used as an SSH client for establishing connections to the HPC. The programming language used is Python 3.10, along with Anaconda, to manage packages and dependencies in both local and remote environments.

6 Results and Discussion

Table 1 summarizes the performance of our submissions in terms of accuracy. Initial experiments with Arabic-specific models, such as Fanar and ALLaM, showed limited effectiveness in handling Islamic inheritance reasoning. The fine-tuned ALLaM model produced similar results. Subsequent submissions explored prompt-based inference with larger multilingual models. The reasoning-focused version of DeepSeek (Deepseek-Reasoner via API) achieved a significant performance gain, and Gemini-2.5-Flash further improved accuracy to 78.10%, high-

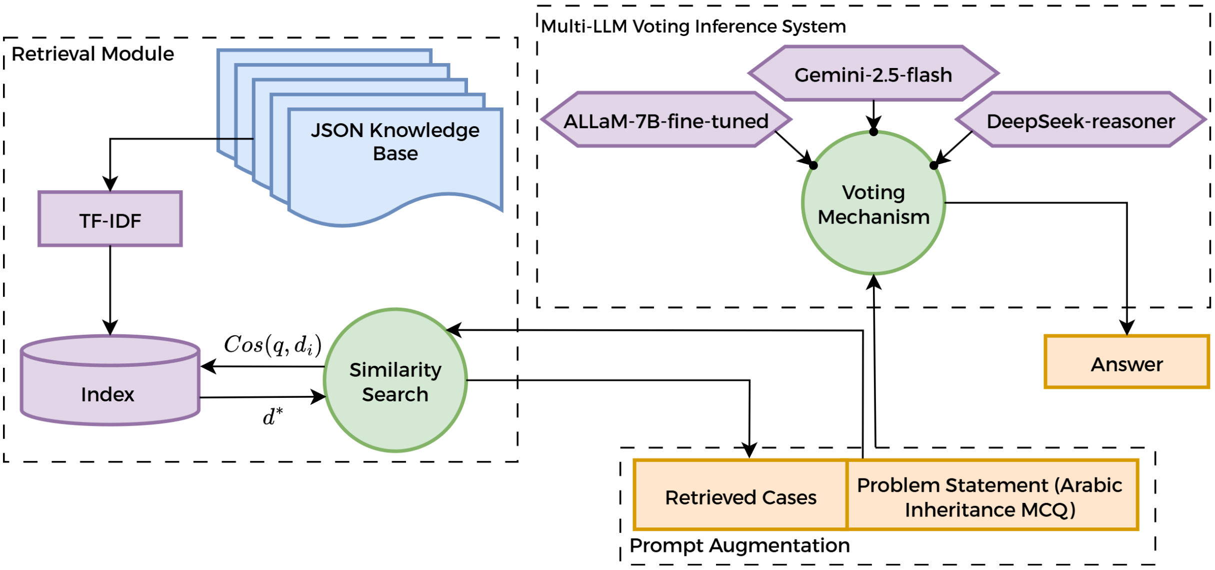


Figure 1: Overall workflow of the implemented collaborative LLMs framework

```
System:
You are a specialist in Islamic sciences. Your task is to answer multiple-choice
questions by selecting the correct option.

User:
Question: {question}

{options_text}

The following fatwas may assist you: {context_text}

Please respond using only one English letter from the following: {valid_letters}
Do not write any explanation or additional text.
```

Figure 2: Prompt structure in the Collaborative LLM Framework; {question} denotes the original inheritance problem statement, {options_text} lists answer choices (one correct), {context_text} contains RAG-retrieved Fatwas for prompt augmentation, and {valid_letters} specifies the expected output format

lighting the benefits of multilingual models over those optimized for a single or limited set of languages. The proposed collaborative voting strategy exploits both the diversity of models (ALLaM, Gemini, DeepSeek), with RAG. This approach achieved the highest accuracy (88.00%), demonstrating the effectiveness of model combination and knowledge retrieval in handling the complex reasoning required for Islamic inheritance cases. A limitation of our voting strategy arises when the three models produce three different answers. In such cases, we default to Gemini's output, given its stronger capability compared to the candidate models. While this rule provided a practical solution in our experiments, it reduces the neutrality of the ensemble. Future work will explore more robust strategies, such as weighted voting, where models' contributions are scaled according to their accuracy.

7 Conclusion

In this work, we introduced a collaborative LLM approach augmented with RAG to tackle Islamic inheritance problems in Arabic. By aggregating heterogeneous models, namely Gemini-2.5-Flash, DeepSeek, and ALLaM-7B, all augmented with RAG, through a majority-vote approach, the system was 88.00% accurate, surpassing the top-performing single model, while achieving stronger robustness to model-specific faults. Our experiments showed that retrieval configuration affects model performance, and TF-IDF at k = 3 performs best, surpassing neural embedding methods under this task. These results report that, for such

Table 1: Accuracy of individual models compared to the collaborative voting approach.

Submission	Model	Accuracy (%)
1	Fanar-1-9B	36.10
2	ALLaM-7B-Instruct-preview	26.50
3	Fine-tuned ALLaM-7B-Instruct-preview	79.50
4	Deepseek-chat	50.90
5	Qwen3-1.7B with RAG	26.10
6	Gemini-2.5-Flash	78.10
7	Gemini-2.5-Flash with RAG	83.50
8	Deepseek-reasoner with RAG	71.80
9	Collaborative Voting	**88.00**

Table 2: Finetuning Hyperparameters

Hyperparameter	Value
Max training steps	300
Batch size	2
Eval batch size	8
Learning rate	5e-5
Max sequence length	1024
Logging frequency	50 steps
Checkpoint frequency	200 steps
Random seed	42
LoRA rank (r)	8
LoRA alpha	16
LoRA dropout	0.05
Optimizer	AdamW
Gradient clipping	1.0
Precision	bfloat16 (GPU)
Max new tokens (eval)	20
Sampling strategy	Greedy

highly structured legal reasoning problems with explicit rules, conventional lexical retrieval can be more successful, and that multi-model collaborative LLMs are more trustworthy in output than solitary models. Further work involves incorporating more extensive and heterogeneous Arabic-specific models, increasing the dataset, and testing the approach for low-resource language translation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alaa N Akkila and Samy S Abu Naser. 2016. Proposed expert system for calculating inheritance in islam.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydın. 2025. Improving llm reliability with rag in religious question-answering: Mufassirqas. *Turkish Journal of Engineering*, 9(3):544–559.

M. Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan Al-Rashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Al-rubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, and 6 others. 2024a. Allam: Large language models for arabic and english. *13th International Conference on Learning Representations, ICLR 2025*, pages 59235–59270.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan Al-Rashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, and 1 others. 2024b. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, ArabicNLP 2025, Suzhou, China, November 5–9, 2025*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Anastasiia Demidova, Hanin Atwany, Nour Rabih, and Sanad Sha'ban. 2024. Arabic train at nadi 2024 shared task: Llms' ability to translate arabic dialects into modern standard arabic. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 729–734.

Md Saef Ullah Miah, Md Mohsin Kabir, Talha Bin Sarwar, Mejdl Safran, Sultan Alfarhood, and MF Mridha. 2024. A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and llm. *Scientific Reports*, 14(1):9603.

Marryam Yahya Mohammed, Sama Ayman Ali, Salma Khaled Ali, Ayad Abdul Majeed, and Ensaf Hussein Mohamed. 2025. Aftina: enhancing stability and preventing hallucination in ai-based islamic fatwa generation using llms and rag. *Neural Computing and Applications*, pages 1–26.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, and 1 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Gumball at QIAS 2025 Shared Task: Arabic LLM Automated Reasoning in Islamic Inheritance

Eman Elrefai*[1], Aml Hassan Esmail*[2], and Mohamed Lotfy Elrefai*[3]

[1]Alexandria University, eman.lotfy.elrefai@gmail.com
[2]Benha University, aml.hassan.esmil@gmail.com
[3]Ain Shams University, mohamed.lotfy.elrefai@gmail.com

Abstract

In this paper, we present a system for solving Islamic inheritance problems using large language models (LLMs), focusing on accurate reasoning in Arabic based on fara'id rules. Our approach is built on the Qwen3-4B model, quantized, and trained using the Unsloth framework for efficiency. We explore multiple training strategies: (1) retrieval-augmented generation (RAG) using fatwas from Islamweb, (2) supervised fine-tuning (SFT) on annotated inheritance datasets, (3) instruction tuning of a base Qwen model followed by GRPO training for multiple choice question solving, and (4) a two-stage pipeline involving SFT on a classical Islamic inheritance book followed by MCQ fine-tuning. Among these, the fourth approach achieved 97.2% accuracy, outperforming all other submissions and ranking our team first in the competition.

1 Introduction

Islamic inheritance laws are complex and highly nuanced, and vary significantly depending on factors such as Islamic sect, national legislation, and cultural practices. Due to this complexity, accurately determining inheritance shares often requires the expertise of scholars well-versed in both jurisprudence and contextual legal systems. This intricate structure makes the domain of Islamic inheritance particularly well-suited for developing reasoning tasks in the Arabic language, offering a rich and challenging environment for natural language understanding and logical inference.

In this work, we present a system that leverages large language models (LLMs) to solve Islamic inheritance problems with high accuracy. We base our approach on the Qwen3-4B (Yang et al., 2025a) model, using the Unsloth framework (Daniel Han and team, 2023) for efficient quantisation and training. To tackle the complexity of the domain, we explore several strategies: retrieval-augmented generation using real-world fatwas, supervised fine-tuning on curated inheritance scenarios, instruction tuning followed by reinforcement training (GRPO) (Shao et al., 2024), and a two-stage pipeline that first fine-tunes on classical texts before solving multiple-choice questions. Our best-performing system as of 2025-08-20, which follows the two-stage pipeline approach, achieved 97.2% accuracy and ranked first on the leaderboard of the Arabic-NLP conference (Bouchekif et al., 2025a,b).

2 Related Work

Prior work in automated Islamic inheritance question answering has been limited, with most systems focusing on rule-based reasoning (Powers, 2017). While these approaches achieve perfect accuracy on explicitly encoded cases, they lack generalisation to unseen problems. Recent advances in Arabic NLP have enabled transformer-based models (Antoun et al., 2020) to tackle domain-specific MCQ tasks, yet most studies address general knowledge or educational exams rather than deep legal reasoning. Our contribution is novel in two aspects: (1) applying a small language model fine-tuned on a large-scale, domain-specific dataset of Islamic inheritance MCQs, and (2) integrating reasoning traces in the training phase (via the reasoning-augmented subset) to improve interpretability and accuracy in complex cases.

3 Dataset

The dataset used in this work consists of Arabic multiple-choice questions (MCQs) in the domain of Islamic heritage. The task involves predicting the correct answer option (A–F) for each question, given six possible choices. This problem requires a combination of reading comprehension, domain-specific legal knowledge, and numerical reasoning.

*These authors contributed equally to this work.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 953–959
November 8-9, 2025 ©2025 Association for Computational Linguistics

The system takes as input a question q in Modern Standard Arabic, typically formulated in formal jurisprudential language, and a set of six possible answer options $\{o_1, o_2, \ldots, o_6\}$. The output is the index of the correct option. For example:

Question:

مات وترك: زوجة (٣) و ابن عم الأب (٥) و بنت
و ابن أخ لأب (٥) و بنت ابن (٥)، كم إجمالي عدد
الأسهم الذي تقسم عليه التركة قبل تصحيح المسألة؟

Options: A. 27 B. 22 C. 25 D. 26 E. 24 F. 23

Answer: E. 24

This setup differs from generic MCQ tasks because the reasoning process often involves applying formal rules from Islamic law, understanding exception cases, and performing share calculations.

3.1 Dataset Splits

The annotated dataset is divided into three splits:

- **Training:** 20,000 questions (10,000 Beginner, 10,000 Advanced)
- **Development:** 1,000 questions (500 Beginner, 500 Advanced)
- **Test:** 1,000 questions (500 Beginner, 500 Advanced, with gold labels for evaluation)

Each question contains six answer options (A–F), exactly one of which is correct. The label distribution in the training set is moderately imbalanced, with option C being the most frequent (21.7%) and option F the least frequent (13.4%) as shown in figure 1 . Table 3 summarises the main statistics.

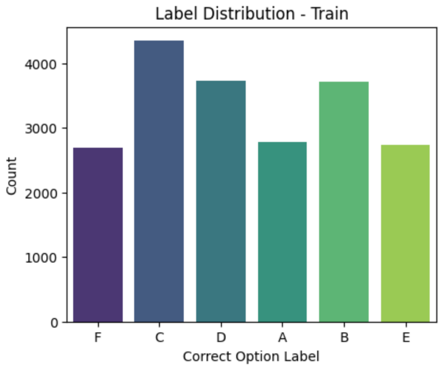


Figure 1: Label distribution in the training set.

3.1.1 IslamWeb Dataset

The IslamWeb corpus contained a total of 3,166 questions distributed across four batches. The dataset was structured as JSON arrays containing detailed fatwa objects with the following fields:

- **ID**: Unique identifier for each fatwa; **URL**: source link on IslamWeb.
- **Category**: Jurisprudential classification.
- **Dates**: Gregorian and Hijri publication dates.
- **Question**: User query; **Answer**: scholar's response with Quran and Hadith references.

3.2 Label and Difficulty Distributions

The label frequencies and difficulty level proportions are illustrated in Figure 1. These reveal a slight imbalance in label frequencies, which may influence model bias toward more frequent options.

This work was conducted in the context of the QIAS 2025– SubTask 1: Islamic Inheritance Reasoning, where participants developed models to predict the correct answer choice. Our submission was evaluated which considers both Beginner and Advanced difficulty levels.

4 System Overview

4.1 Two-Stage Fine-Tuning of SLM (Continual Pretraining + SFT)

Our end-to-end Figure 2 is organised as three distinct stages that are executed in a pipeline:

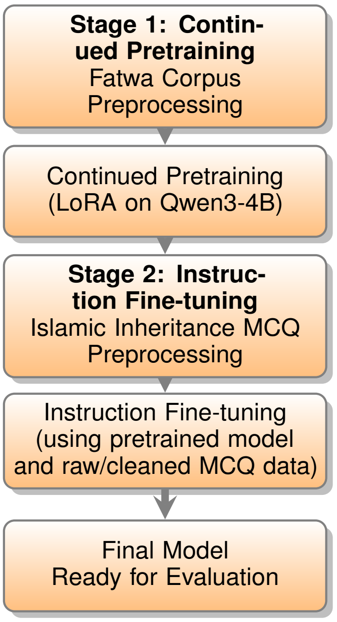

Stage 1: Continued Pretraining
Fatwa Corpus Preprocessing
Continued Pretraining (LoRA on Qwen3-4B)
Stage 2: Instruction Fine-tuning
Islamic Inheritance MCQ Preprocessing
Instruction Fine-tuning (using pretrained model and raw/cleaned MCQ data)
Final Model Ready for Evaluation

Figure 2: Two-Stage Fine-Tuning Pipeline for Islamic Legal Text Modelling.

1. **Domain Continual Pretraining:** We perform LoRA-based (Hu et al., 2022) continual pretraining of a Qwen3 family base model on a curated IslamWeb fatwa/article corpus to adapt the model to jurisprudential registers, domain phrases, and common reasoning patterns. The pretraining objective is standard autoregressive next-token likelihood.

2. **Supervised Fine-Tuning (SFT).** We fine-tune the adapted model on the MCQ inheritance dataset using instruction-style prompts (question + choices → answer token or short explanation). SFT enforces the mapping from the problem statement to the correct option and optionally to an intermediate reasoning trace.

3. **Cleanup & Post-processing.** A lightweight script normalises Arabic diacritics (tashkeel), punctuation, and simple orthographic variants – both as a preprocessing step for training and as a post-processing step on model outputs prior to scoring. This reduces spurious surface mismatches between model outputs and gold labels.

4.2 Reinforcement Learning Fine-Tuning

We fine-tuned the base `Qwen3-4B` model on an Islamic inheritance reasoning dataset using supervised fine-tuning (SFT) with instruction-style prompts, where each input was a question and the output contained step-by-step reasoning.

Following the DeepSeek reasoning framework (Shao et al., 2024), we applied reinforcement learning fine-tuning (RLFT) with the GRPO algorithm, training the model to produce both detailed reasoning traces and final multiple-choice answers.

To guide RLFT, we implemented the following custom reward functions:

- **Template Matching (Exact/Approximate)** – enforce reasoning and solution structure using predefined tokens.

- **Answer Format Validation** – ensure answers match valid multiple-choice options.

- **Numerical Accuracy Check** – reward exact matches to ground truth values.

- **Fuzzy Matching** – grant partial credit for near-correct outputs in format or structure.

These rewards balanced structural consistency, factual accuracy, and reasoning quality during training.

4.3 Retrieval-Augmented Generation

We implemented a Retrieval-Augmented Generation (RAG) system as an initial baseline, combining the competition-provided domain-specific corpus with additional external resources to expand coverage and improve retrieval quality.

For the retrieval component, we employed dense vector embeddings and evaluated several multilingual models: `e5-base` (Wang et al., 2024), `MiniLM-L12-v2` (Reimers and Gurevych, 2019), and `Matryoshka` (Nacar and Koubaa, 2024). Among these, the `Matryoshka` model consistently achieved the highest retrieval accuracy in our experiments.

The generation component was powered by `Qwen2.5-7B` (Yang et al., 2025b) using the Ollama v0.11.10 (Ollama Team, 2023) inference framework.

5 Experimental Setup

5.1 Two-Stage Fine-Tuning of SLM Pipeline

Training configuration and hyperparameters: The key training hyperparameters used across experiments are listed in Table 5 is provided in the Appendix.

5.1.1 Two-Stage Fine-Tuning Prompt

We used the following prompt format for training the first stage:

```
### الموضوع: {}
### السؤال:{}
<think>
### الإجابة:{}
</think>
```

Second stage multiple-choice questions, we formatted the prompts as:

System Prompt:

أنت مساعد ذكي تتحدث باللغة العربية الفصحى.

كن مهنيًا، لبقا وودودا في ردودك.

أجب بوضوح وتفصيل، وتجنب الردود المختصرة.

User Prompt:

"ما هي الإجابة الصحيحة على السؤال التالي؟
.\n\n"
"السؤال: {example['question']}\n"

```
الاختيارات :
"A. {example['option1']}\n"
"B. {example['option2']}\n"
"C. {example['option3']}\n"
"D. {example['option4']}\n"
"E. {example['option5']}\n"
"F. {example['option6']}\n"
جاوب برمز الاجابة الصحيحة فقط
```

5.1.2 Hardware Configuration

Experiments were conducted on a system equipped with an NVIDIA GeForce RTX 3090 Ti GPU with 24GB VRAM. This hardware provided sufficient memory for efficient training of the 4B parameter models using the Unsloth framework with LoRA fine-tuning.

5.2 Reasoning Pipeline

We fine-tuned the `Qwen3-4B-Base` model using the `Unsloth` framework for efficient training and inference. The model was configured with a 2048-token context window and LoRA rank 32 applied to projection and feed-forward layers. Gradient checkpointing and a 70% GPU memory cap were used to reduce resource usage.

5.2.1 Supervised Fine-Tuning

SFT was performed for 2 epochs with batch size 1 using the AdamW (Loshchilov and Hutter, 2017) 8-bit optimizer, learning rate $2e^{-4}$, weight decay 0.01, and linear scheduling with 5 warm-up steps.

5.2.2 Reinforcement Learning Fine-Tuning

We then applied (GRPO) with `vllm` for fast sampling. Settings included 4 generations per prompt, temperature 1.0, `top_p` = 1.0, and learning rate $5e^{-6}$ for 100 steps. Multiple reward functions were used to enforce output format and correctness.

5.3 RAG Pipeline

5.3.1 Data Sources & Preprocessing

We combined the corpus provided by the competition with external inheritance resources. JSON files were converted into structured Q&A pairs, while unstructured documents were segmented using two strategies:

- Q&A extraction: regex-based identification of question–answer patterns.
- Semantic chunking: splitting long passages into 400-token segments with guiding questions.

All text was normalized through diacritic removal, character unification, stopword filtering, and whitespace cleanup, reducing noise and improving retrieval quality.

5.3.2 Retrieval

Documents were embedded using dense vector models from the SentenceTransformers library. We evaluated `e5-base`, `MiniLM-L12-v2`, and `Matryoshka`.

The last of these achieved the best retrieval accuracy in our domain. Retrieval employed cosine similarity, and for the best results, we used a top-3 selection strategy and a minimum similarity threshold of 0.7.

6 Results

6.1 Two-Stage Fine-Tuning of SLM Pipeline

6.1.1 Main results

Table 8 summarises the most relevant submissions (sorted by test accuracy). For each run, we report whether the cleanup script was applied (preprocessing and/or post-processing), the development accuracy (noting whether the dev split was cleaned), and the test accuracy used in the leaderboard submission.

6.1.2 Ablation: cleanup vs. no-cleanup

We compare matched runs where the only difference is whether the evaluation is performed on cleaned or raw data. The most illustrative matched pair is experiment F (raw training, evaluated on the cleaned test set) versus experiment H (raw training, evaluated on the raw test set):

- **Exp F (Raw $\rightarrow$ Clean Test):** Test 95.1%.
- **Exp H (Raw $\rightarrow$ Raw Test):** Test 94.3%.

This indicates that applying the deterministic cleanup procedure during evaluation yields a measurable improvement in final test accuracy (+0.8 percentage points in this pair).

6.2 Reasoning Pipeline

- **Baseline RLFT performance:** Applying RLFT directly to the `Qwen3-4B` model yielded **15%** accuracy.
- **Domain-adapted initialisation :** Initialising RLFT (500 steps) from a checkpoint fine-tuned on the Islamic inheritance MCQ dataset achieved **57%** accuracy.

Table 1: Accuracy of Different Inheritance Reasoning Pipelines on the test dataset

Pipeline	Accuracy (%)
RAG (Qwen2.5-7B + best embedding model)	35.33
Instruction SFT + GRPO	57.00
SFT on Annotated Dataset	87.00
Two-Stage Fine-Tuning of SLM (Continual Pretraining + SFT)	**97.20**

These results highlight the advantage of starting from a domain-adapted model for improving reasoning performance. Further optimisation of RLFT was not pursued due to time and resource constraints.

6.3 RAG Pipeline

We conducted a two-stage evaluation process on the development dataset:

1. **Pre-RAG Evaluation:** As shown in Table 2, `Qwen2.5-7B` (Yang et al., 2025b) achieved the highest standalone accuracy (31.5%), clearly outperforming both `Qwen3-4B` and `Qwen3-8B`. This established it as the strongest baseline model prior to retrieval integration.

2. **RAG Integration:** Building on this superior baseline, we integrated `Qwen2.5-7B` with our retrieval pipeline. Table 4 shows that combining the model with different embedding backbones led to further improvements, with the best accuracy (44.0%) obtained using the `Arabic-all-nli-triplet-Matryoshka` embeddings.

The RAG pipeline delivered an absolute gain of 12.5% (about 39.7% relative) over the standalone baseline, highlighting the value of targeted retrieval in knowledge-intensive tasks. While it did not surpass our fine-tuned models, it remains a strong, resource-efficient option for settings with limited computational budgets.

7 Analysis of Result

The model demonstrates strong performance overall, but a detailed analysis of its failures is crucial for future improvements. The overall error rate is low, though it is notably higher for questions categorised as “Advanced” (5.0%) compared to those labelled “Beginner” (0.6%). This suggests that the model struggles more with complex inheritance scenarios. Such performance gaps align with the concerns raised by (Fawzi et al., 2025; Sibaee et al., 2025), who highlight that errors in large language models in Arabic and religious contexts, particularly in complex reasoning tasks, can have serious consequences. The following tables 6 and 7 present representative failure cases, followed by a detailed analysis of the underlying reasons for the incorrect predictions.

7.1 Statistics

The model was evaluated on 1000 test questions, equally split between ’Beginner’ and ’Advanced’ levels. Overall accuracy was 97.2%, with a total error rate of 2.8%. Errors were more frequent in ’Advanced’ questions (5.0%) compared to ’Beginner’ ones (0.6%), indicating strong performance on basic rules but reduced accuracy in complex cases involving multiple heirs, distant kinship, and share correction (*tas’hih*).

8 Conclusion

We built an Arabic system for solving Islamic inheritance problems using large language models, achieving first place in QIAS 2025 with 97.2% accuracy. Our two-stage fine-tuning—domain continual pretraining plus supervised fine-tuning—was most effective, aided by targeted preprocessing. While basic cases were nearly flawless, complex scenarios require improvements in reasoning, symbolic integration, and interpretability for reliable real-world use.
For reproducibility, the implementation and code are available at `Gumball at QIAS 2025 | GitHub`.

9 Acknowledgments

We thank the QIAS 2025 shared task organisers for providing this valuable evaluation framework. We also acknowledge the anonymous reviewers for their constructive feedback.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, ArabicNLP 2025, Suzhou, China, November 5–9*. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Mahmoud Fawzi, Björn Ross, and Walid Magdy. 2025. Fabricating holiness: Characterizing religious misinformation circulators on arabic social media. *Preprint*, arXiv:2508.07845.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu et al. Chen. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Omer Nacar and Anis Koubaa. 2024. Enhancing semantic similarity understanding in arabic nlp with nested embedding learning. *Preprint*, arXiv:2407.21139.

Ollama Team. 2023. Ollama: An open-source framework for local llm inference. `https://github.com/ollama/ollama`. Accessed: Aug 1, 2025.

David S Powers. 2017. The islamic inheritance system: a socio-historical approach. In *Issues in Islamic Law*, pages 165–181. Routledge.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and Yang et al. Wu. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Serry Sibaee, Omer Nacar, Adel Ammar, Yasser Al-Habashi, Abdulrahman Al-Batati, and Wadii Boulila. 2025. From guidelines to practice: A new paradigm for arabic language model evaluation. *Preprint*, arXiv:2506.01920.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, and Chenxu et al. Lv. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, and Jingren et al. Zhou. 2025b. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*.

A Appendix

1.1 Tables

Table 2: Performance of different base Qwen Models on the development dataset

Model	Accuracy (%)
Qwen3-4B	10.8
Qwen3-8B	15.0
Qwen2.5-7B	**31.5**

Table 3: Dataset statistics by split.

Split	# Questions	Beginner	Advanced
Train	20,000	10,000	10,000
Dev	1,000	500	500
Test	1,000	500	500

Table 4: Performance of different embedding models on the development dataset

LLM Model	Embedding Model	Accuracy (%)
Qwen2.5-7B	paraphrase-multilingual-MiniLM-L12-v2	38.0
	multilingual-e5-base	42.6
	Arabic-all-nli-triplet-Matryoshka	**44.0**

Table 5: Key hyperparameters (representative values).

Hyperparameter	Pretraining	SFT
Base model	Qwen3-4B	PreTrain Model Qwen3 with LoRA
LoRA rank (r)	128	128
LoRA α	16	16
Context length	2048	2048
Batch size (per GPU)	12 (accumulation)	12 (accumulation)
Optimizer	AdamW	AdamW
Learning rate	5e−5	5e−5
Embedding Learning rate	1e−5	1e−5
Warmup steps	5	5
Weight decay	0.01	0.00
Epochs	3	4 (SFT)
Precision	bf16	bf16

Table 6: Analysis of Failure Case 1: Complex Kinship

Failure Case 1: Complex Kinship	
ID	`4232_nq7p3f6g_18`
Level	Advanced
Question	مات وترك: أم أب الأب و ابن ابن أخ لأب (٢) الأم و عم شقيق (٣) و، وام كم عدد الأسهم بعد التصحيح التي يحصل عليها لكل ابن ابن أخ لأب؟
Prediction	A: 3 shares
Ground Truth	F: 1 share
Analysis	**Key Error**: Miscalculated *tas'hih* (correction) for agnatic heirs **Reason**: Incorrect priority order determination **Fix**: Improve share correction logic

Table 7: Analysis of Failure Case 2: Exclusion Error

Failure Case 2: Exclusion Error	
ID	`1981_nm1l6g8b_1`
Level	Advanced
Question	مات وترك: أم الأم و أخ لأب و ابن عم شقيق (٣) و ابن (٥) و أب الأب كم النصيب الأصلي لكل صنف من الورثة من التركة؟
Prediction	B: أم الأم: السدس، أب الأب: محجوب، ...
Ground Truth	F: أم الأم: السدس، أب الأب: السدس، ...
Analysis	**Key Error**: Excluded grandfather **Reason**: Core rule misunderstanding **Fix**: Correct exclusion principles

Table 8: Two-Stage Fine-Tuning of SLM pipeline: results with cleaned vs. raw training data. **Base model:** Qwen pretrained on IslamWeb.

Exp	Data	Pre Steps	Cleanup Steps	Eval Set	Dev (%)	Test (%)
A	Cleaned	5500	3000	Clean	81.9	97.2
B	Cleaned	5500	4500	Clean	82.4	97.0
C	Cleaned	5500	4834	Clean	82.5	96.8
D	Cleaned	5500	2500	Clean	82.0	96.8
E	Cleaned	5500	1500	Clean	80.7	96.4
F	Raw	2500	–	Clean	–	95.1
G	Raw	5500	–	Raw	80.7	95.8
H	Raw	2500	–	Raw	78.9	94.3

Tokenizers United at QIAS 2025 Shared Task: A Retrieval-Augmented Generation Pipeline for Islamic Knowledge Assessment

Mohamed Samy[2], Mayar Boghdady[1], Marwan El Adawi[1], Mohamed Nassar[1], Ensaf Hussein[1]
[1] School of Information Technology and Computer Science, Nile University
[2] Faculty of Computer and Information Sciences, Ain Shams University
MBoghdadi@nu.edu.eg, M.Mahmoud2179@nu.edu.eg, M.Ali2265@nu.edu.eg, EnMohamed@nu.edu.eg
mohamedsamyy02@gmail.com

Abstract

This paper presents the approach and results for Sub Task 2: General Islamic Knowledge Question Answering at QIAS 2025, a shared task designed to evaluate the capabilities of Large Language Models (LLMs) in answering multiple-choice questions across diverse domains of Islamic knowledge, including theology, jurisprudence, biography, and ethics. A Retrieval-Augmented Generation (RAG) system powered by the Gemini language model was developed for this task.

In the proposed system, the *retriever module* performs semantic search over curated classical Islamic sources to identify passages relevant to each input question, while the *generator module* leverages the LLM to reason over the retrieved evidence and generate a final answer. This integration of evidence retrieval with contextual reasoning enables accurate responses across diverse knowledge areas.

On the official test set, the system achieved an accuracy of 87%, ranking 5th out of 10 participating teams in QIAS 2025 Sub Task 2. These results demonstrate the effectiveness of combining retrieval-based evidence with generative reasoning in specialized religious domains, highlighting the potential of RAG architectures for *high-stakes, knowledge-intensive question answering tasks* and confirming their robustness in the QIAS 2025 benchmark.

1 Introduction

Automated assessment of Islamic knowledge is a critical task requiring both linguistic proficiency and deep domain expertise. It faces challenges from the complexity of Arabic morphology and orthography, the breadth of Islamic sources, and the demand for trustworthy responses in educational contexts.

Within the context of the QIAS2025 Shared Task (Sub Task 2: Islamic Assesment) , existing methods based on generic LLMs, classical retrieval, or translation pipelines often fail to capture domain-specific semantics, suffer from hallucinations, and lack grounding in authoritative sources. This highlights a gap between current capabilities and the requirements of knowledge- intensive domains such as Islamic studies.

To address this, a Retrieval-Augmented Generation (RAG) framework is proposed, combining Muffakir embeddings for evidence retrieval with Gemini 2.5 Flash Lite for generative reasoning. Preprocessed texts are segmented into enriched units for efficient retrieval, ensuring grounded and accurate responses. Experiments show the system achieves 84% precision on development data and 87% on the official test set, outperforming baselines in the QIAS 2025 evaluation (Bouchekif et al., 2025a).

The main contributions are: (1) a curated Arabic knowledge base for Islamic studies, (2) integration of retrieval with a state-of-the-art LLM, and (3) empirical validation in high-stakes assessment tasks under the QIAS 2025 benchmark (Bouchekif et al., 2025a).

2 Related Work

Several studies have explored the development of Islamic Question Answering (QA) systems, following either retrieval-based methods or knowledge-based approaches enhanced with semantic processing. An early example is (Mohamed et al., 2015), which introduced *Al-Bayan*, a knowledge-based Arabic answer selection system for Islamic sciences that participated in SemEval-2015 Task 3. By combining a Quranic ontology enriched with Tafseer resources, keyword matching, and a decision tree classifier, the system achieved an accuracy of 74.53% and a macro-F1 score of 67.65%.

A broader perspective is provided in (Alnefaie

Proceedings of The Third Arabic Natural Language Processing Conference, pages 960–965
November 8-9, 2025 ©2025 Association for Computational Linguistics

Question	Answers	Level	Label
ما هو القول القديم للشافعي في صوم أيام التشريق؟	A) لا يجوز صومها مطلقاً. B) يجوز صومها للمتمتع إذا عدم الهدي عن الأيام الثلاثة الواجبة في الحج. C) يجوز صومها لمن لم يجد الهدي فقط. D) يجوز صومها للمسافر فقط.	advanced	B
أنا سورة قصيرة، نزلتُ كاملة بسبب كلمة غضب قالها قريب للنبي -صلى الله عليه وسلم- رداً على دعوته. فما اسمي؟	A) سورة الكافرون B) سورة المسد (تبت) C) سورة النصر D) سورة الهمزة	beginner	B
أنا اختلاف ألفاظ الوحي المنزل في الحروف أو كيفيتها من تخفيف وتشديد وغير ذلك. فماذا أكون حسب تعريف الزركشي؟	A) التجويد B) القرآن C) أسباب النزول D) القراءات	advanced	D
بماذا عرفت أخت أنس بن النضر أخاها الذي استشهد يوم أحد؟	A) بوجهه B) بيده C) ببناته D) كل الأجوبة خطأ	intermediate	C
اختر الآية من سورة الضحى التي تدل على معنى عمما صرمك فتركك، وما أبغضك منذ أحبك"	A) ألم يجدك يتيماًا فآوى B) ووجدك ضالاًا فهدى C) ووجدك عائلاًا فأغنى D) ما ودعك ربك وما قلى	intermediate	D

Table 1: Sample of Islamic knowledge assessment questions with answer options (A–D), difficulty level, and correct label.Latin labels are forced with `\textlatin{}` to avoid RTL localization.

et al., 2023), which presented a comprehensive survey of Islamic QA systems drawing on Qur'an, Hadith, and Fatwa sources. Their evaluation classified systems into traditional retrieval-based and knowledge-based categories, with deep learning models such as AraBERT, AraElectra, and mT5 showing promise but remaining highly dependent on dataset quality. The survey applied thirteen evaluation criteria, concluding that most current systems suffer from limited coverage, lack of public availability, and difficulty in handling non-factoid questions.

Recent contributions have expanded the scope of Islamic QA to general knowledge domains including theology, jurisprudence, biography, and ethics. (Qamar et al., 2024) introduced a large-context Islamic QA dataset for non-factoid questions, derived from Qur'an, Tafsir, and Hadith. Domain-specific legal reasoning has also been addressed; for example, (Al-Qurishi et al., 2022) proposed *AraLegal-BERT*, a BERT model fine-tuned on Arabic legal texts to enhance QA in Islamic jurisprudence. Other benchmarks include (Malhas, 2023), which developed *QuranQA* for span selection tasks, and (Premasiri et al., 2022), which introduced *MadinaQA* for beginner and intermediate Islamic studies. Advances in Retrieval-Augmented Generation (RAG) were demonstrated by (Alan et al., 2024), who presented *MufassirQAS*, while (Rizqullah et al., 2023) proposed *QASiNa*, targeting QA over Sirah Nabawiyah texts.

Work has also extended beyond Arabic into Persian, with (Ghafouri et al., 2023), (Etezadi and Shamsfard, 2021), and (Zeinalipour et al., 2025) developing QA systems and benchmarks for multi-hop and multiple-choice reasoning. Domain-specific applications include Islamic inheritance law, where (Bouchekif et al., 2025b) provided benchmarks and evaluations of large language models for legal reasoning.

Taken together, these studies highlight the increasing interest in combining knowledge-based and deep learning approaches to address the challenges of Islamic QA. The literature underscores the importance of multilingual support, robust reasoning across complex religious texts, and domain-specific legal knowledge representation, while also pointing to the potential of modern language models and Retrieval-Augmented Generation to advance the field.

2.1 Task Setup: QIAS 2025 Shared Task

The (Bouchekif et al., 2025a) shared task has been established as a benchmark competition to evaluate systems for Islamic Question Answering. It consists of multiple subtasks designed to assess models in handling diverse domains of Islamic knowledge. This work focuses on **(Bouchekif et al., 2025a) Subtask 2: General Islamic Knowledge QA**, which targets multiple-choice question answering across domains including theology, jurisprudence, biography, and ethics.

Subtask 2 attracted participation from ten international teams. Evaluation was based on system accuracy in selecting the correct option among four candidates. The system presented in this paper ranked **5th out of 10** with an accuracy of 0.875, demonstrating the competitiveness of lightweight RAG pipelines against more complex architectures.

2.2 Dataset

To evaluate the proposed approach, the **QIAS 2025** (Bouchekif et al., 2025a) dataset is used. This benchmark includes multiple-choice questions on *Qur'anic studies, Hadith, Fiqh, Islamic history, and Arabic linguistics*, each annotated with difficulty level (*beginner, intermediate, advanced*) and the correct label. The dataset covers both factual recall and higher-order reasoning, enabling assessment of comprehension and semantic interpretation in Islamic knowledge. Table 1 shows sample questions with answer options, difficulty levels, and correct labels, highlighting the diversity of jurisprudential, exegetical, and historical content.

In addition to the question–answer pairs, the QIAS organizers provide a collection of classical Islamic reference works that serve as the textual backbone for knowledge-intensive tasks. These include:

- *Usul al-Fiqh and Legal Maxims*
- *Al-Itqan fi Ulum al-Qur'an* (The Perfect Guide to the Sciences of the Qur'an)
- *Al-Sirah wa al-Shama'il* (Prophetic Biography and Characteristics)
- *Tashnif al-Masamih bi-Jam' al-Jawami'* (A Comprehensive Collection of Jurisprudential Principles)
- *Manhaj al-Naqd fi Ulum al-Hadith* (Methodology of Criticism in the Sciences of Hadith)

These books were segmented into smaller chunks and indexed to form the system's knowledge base, allowing retrieval-augmented generation to ground answers in authoritative Islamic sources.

3 Methodology

The Islamic knowledge assessment system is designed as a multistage pipeline that combines Arabic text preprocessing, embedding-based retrieval, and large language model (LLM) generative reasoning. The overall workflow is depicted in Figure 1, which describes the stages from raw document ingestion to final response generation. The system architecture begins with a query question, followed by query embedding, vector search for relevant knowledge chunks, prompt construction, LLM-based reasoning, and ultimately the production of an answer.

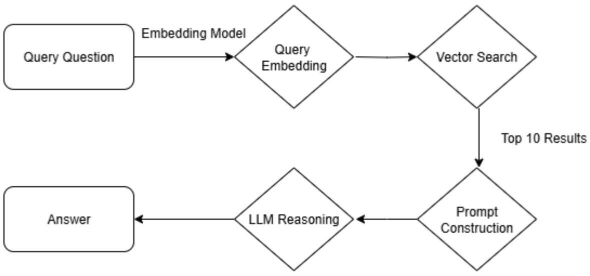


Figure 1: Proposed pipeline of the Islamic knowledge assessment system.

Document ingestion is performed by extracting text from diverse file formats. The extracted content undergoes cleaning, which involves the removal of diacritics, Tatweel, unwanted symbols, phone numbers, emails, and URLs. Text normalization for punctuation and whitespace is applied to ensure consistency. After cleaning, the text is split into overlapping chunks, which may be based on words, sentences, or paragraphs. Each chunk is annotated with metadata such as keywords, positional information, and unique identifiers.

Embedding-based retrieval constitutes the second stage of the pipeline. The preprocessed text chunks are transformed into dense vector representations using the `Muffakir Embedding` model. These embeddings are stored in a vector database, allowing efficient similarity-based retrieval when a query is introduced. The user query is also converted into an embedding, which is compared against the stored vectors to identify the most semantically relevant passages.

Prompt construction serves as the bridge between retrieval and reasoning. Once relevant chunks are retrieved, they are assembled together with the user query into a structured prompt. This ensures that the LLM receives not only the query but also the most contextually aligned passages, enabling grounded and accurate responses.

Generative reasoning is executed by large language models that process the constructed prompt. Several models were evaluated, including Silma, Qwen3 1.7B/8B, Aya, and Allam. Among these, Gemini 2.5 Flash Lite demonstrated superior performance in generating coherent and contextually faithful answers. A flash reranker was also tested for post-retrieval refinement; however, direct retrieval combined with Gemini exhibited more reliable outcomes.

The final architecture integrates these components into a Retrieval-Augmented Generation (RAG) system. Beginning with the query question, the process proceeds through query embedding, vector search, prompt construction, and LLM-based reasoning, which culminates in the generation of the final answer. This structured pipeline enables the system to leverage external knowledge repositories while preserving the fluency and reasoning ability of modern LLMs. Table 1 provides illustrative examples of multiple choice questions (MCQs) produced by the system, including their correct labels and difficulty levels.

4 Results

in Subtask 2 (Accuracy: 0.93). Detailed scores and rankings are shown in Table 5 (Appendix B As part of the QIAS 2025 Shared Task (Bouchekif et al., 2025a), this system was evaluated on Subtask 2. On the development set, accuracies ranged between 44.29% and 84.29% across different configurations (Table 2). The highest score (84.29%) was obtained using Gemini 2.5 Flash Lite with Muffakir_Embedding and direct similarity search (Top-K = 10), showing that lightweight, well-aligned components can surpass more complex pipelines.

Model size did not consistently translate into better results, as the Qwen3 models showed variable performance across scales (54.43–78.00%). Embedding choice was the most decisive factor: Muffakir_Embedding consistently outperformed other embeddings, while silma-embedding-matryoshka-v0.1 achieved the lowest accuracy (44.29%). Retrieval strategy also proved critical, with direct retrieval outperforming reranking approaches (Flash, BGE). Chain-of-thought prompting offered only modest improvements compared to embedding and retrieval methods.

Overall, the findings highlight that domain-specific embeddings, lightweight LLMs, and simple retrieval mechanisms are more effective than scaling models or adding complex reasoning layers. Our optimized configuration—Qdrant with cosine similarity, a chunk size of 400 characters with overlap of 100 characters, Top-10 retrieval, and 768-dimensional Muffakir_Embeddings—achieved 87.00% accuracy on the held-out test set, confirming strong generalization. The complete set of hyperparameters for this configuration is summarized in (Table 3).

A breakdown by difficulty level (Table 4) shows that performance was highest on beginner questions (89.14%), followed by intermediate (83.43%), while advanced questions proved more challenging (75.43%). This suggests that while the system is robust in handling straightforward queries, further optimization is needed to improve reasoning in complex or nuanced scenarios.

5 Conclusion

Within the framework of the QIAS 2025 (Bouchekif et al., 2025a) Shared Task , specifically Subtask 2, this study demonstrates that effective automated assessment in the domain of Islamic knowledge can be achieved through a carefully optimized Retrieval-Augmented Generation (RAG) pipeline. The experiments confirm that domain-specific embeddings—particularly Muffakir_Embedding—when paired with a lightweight yet capable LLM such as Gemini 2.5 Flash Lite, significantly outperform larger, general-purpose models. Contrary to common assumptions, complex reranking strategies and large-scale models did not yield superior results; in this case, direct retrieval with cosine similarity achieved the highest accuracy of 87%.

The findings underscore three key lessons for high-stakes, domain-specific QA systems: (1) high-quality, domain-tuned embeddings are critical for precision, (2) retrieval quality has a greater impact than advanced reasoning prompts or reranking layers, and (3) computational efficiency and scalability can be maintained without sacrificing accuracy.

In the official results of the shared task, the system achieved a ranking of **5th out of 10 teams** in (Bouchekif et al., 2025a) Subtask 2, highlighting

Table 2: Development set accuracies across configurations. “Reranker” indicates post-retrieval reranking. “CoT” indicates Chain-of-Thought prompting.

LLM Model	Embedding model	Reranker / CoT	Acc (%)
Qwen3 (0.6B)	Arabic-Triplet-Matryoshka-V2	-	54.4
Qwen3 (1.7B)	Arabic-Triplet-Matryoshka-V2	-	62.4
Gemini 2.5 Flash Lite	**Muffakir_Embedding**	**-**	**84.3**
Gemini 2.5 Flash Lite	Muffakir_Embedding	Flash reranker	77.9
Qwen3 (8B)	mohamed2811/Muffakir_Embedding	-	58.4
Aya (8B)	silma-embedding-matryoshka-v0.1	-	44.3
Qwen3 (8B)	Muffakir_Embedding	-	78.00
Qwen3 (8B)	silma-embedding-matryoshka-v0.1	BGE-reranker-v2-m3	69.0
Qwen3 (8B)	mohamed2811/Muffakir_Embedding	BGE-reranker-v2-m3 + CoT	75.0

Table 3: Key hyperparameters (final configuration).

Component	Value
Chunk size	400 characters
Overlap	100 characters
Top-K retrieval	10 (based on cosine similarity)
Embedding dim	768 (Muffakir_Embedding)
Vector count	∼15,000
HNSW: m	64
HNSW: ef_construct	1024
HNSW: full_scan_threshold	0
HNSW: payload_m	96
Optimizers: indexing_threshold	14,000
Optimizers: default_segment_number	40
Optimizers: max_optimization_threads	4

Table 4: Accuracy by difficulty level.

Level	Wrong	Correct	Accuracy (%)
Advanced	43	132	75.43
Beginner	38	312	89.14
Intermediate	29	146	83.43

its competitiveness in a multilingual and domain-sensitive evaluation setting. Future research directions include expanding the knowledge base to additional Islamic sciences, incorporating multilingual capabilities for cross-lingual assessment, and integrating adaptive difficulty calibration to further enhance learner evaluation.

Limitations

A primary limitation of this work lies in the restricted computational resources, which prevented extensive experimentation with larger and more advanced reasoning models that could potentially achieve higher accuracy. In addition, the evaluation of larger and more precise embedding models, as well as the use of computationally intensive reranking strategies, was not feasible under the available setup. These constraints may have capped the system’s performance ceiling, suggesting that future studies with greater resources could further enhance both retrieval quality and answer generation.

Acknowledgments

The organizers of QIAS 2025: Islamic Q&A Shared Task are gratefully acknowledged for providing the datasets and guidance that enabled this work. The QIAS benchmark,(Bouchekif et al., 2025a) covering general Islamic knowledge, served as the basis for system development and evaluation. Appreciation is also extended to the scholars and annotators for curating and verifying the bilingual datasets, and to the Association for Computational Linguistics (ACL) for offering a platform to disseminate this research.

References

Muhammad Al-Qurishi, Sarah AlQaseemi, and Riad Soussi. 2022. Aralegal-bert: A pretrained language model for arabic legal text. *arXiv preprint arXiv:2210.08284*.

Ahmet Yusuf Alan, Enis Karaarslan, and Ömer Aydin. 2024. A rag-based question answering system proposal for understanding islam: Mufassirqas llm. *CoRR*, abs/2401.15378.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Islamic question answering systems survey and evaluation criteria. *International Journal on Islamic Applications in Computer Science and Technology*, 11(1):9–18.

Abdessalam Bouchekif, Samer Rashwani, Mohammed Ghaly, Mutaz Al-Khatib, Emad Mohamed, Wajdi Zaghouani, Heba Sbahi, Shahd Gaben, and Aiman Erbad. 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, Ara-*

bicNLP 2025, Suzhou, China, November 5–9, 2025. Association for Computational Linguistics.

Abdessalam Bouchekif, 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 (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics.

Romina Etezadi and Mehrnoush Shamsfard. 2021. A knowledge-based approach for answering complex questions in persian. *CoRR*, abs/2107.02040.

Arash Ghafouri, Hasan Naderi, Mohammad Aghajani Asl, and Mahdi Firouzmandi. 2023. Islamicpcqa: A dataset for persian multi-hop complex question answering in islamic text resources. *CoRR*, abs/2304.11664.

Rana Malhas. 2023. Fine-tuning arabic qa models for qur'an qa task. In *Proceedings of the 2022 Workshop on Open-Source Arabic Corpora and Tools (OSACT 2022)*.

Reham Mohamed, Maha Ragab, Heba Abdelnasser, Nagwa M. El-Makky, and Marwan Torki. 2015. Albayan: A knowledge-based system for arabic answer selection. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 226–230. Association for Computational Linguistics.

Damith Premasiri, Tharindu Ranasinghe, Wajdi Zaghouani, and Ruslan Mitkov. 2022. Dtw at qur'an qa 2022: Utilizing transfer learning with transformers for question answering in a low-resource domain. In *Proceedings of the Qur'an QA 2022 Shared Task*.

Faiza Qamar, Seemab Latif, and Rabia Latif. 2024. A benchmark dataset with larger context for non-factoid question answering over islamic text. *arXiv preprint arXiv:2409.09844*.

Muhammad Razif Rizqullah, Ayu Purwarianti, and Alham Fikri Aji. 2023. Qasina: Religious domain question answering using sirah nabawiyah. In *arXiv preprint*.

Kamyar Zeinalipour, Neda Jamshidi, Fahimeh Akbari, Marco Maggini, Monica Bianchini, and Marco Gori. 2025. Persianmcq-instruct: A comprehensive resource for generating multiple-choice questions in persian. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages (LoResLM)*, pages 344–372. Association for Computational Linguistics.

TAQEEM 2025: Overview of The First Shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions

May Bashendy
Qatar University
ma1403845@qu.edu.qa

Salam Albatarni
Qatar University
sa1800633@qu.edu.qa

Sohaila Eltanbouly
Qatar University
se1403101@qu.edu.qa

Walid Massoud
Qatar University
wmassoud@qu.edu.qa

Houda Bouamor
Carnegie Mellon University in Qatar
hbouamor@cmu.edu

Tamer Elsayed
Qatar University
telsayed@qu.edu.qa

Abstract

Automated Essay Scoring (AES) has emerged as a significant research problem in natural language processing, offering valuable tools to support educators in assessing student writing. Motivated by the growing need for reliable Arabic AES systems, we organized the first shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions (*TAQEEM*) held at the ArabicNLP 2025 conference. *TAQEEM* 2025 includes two subtasks: Task A on holistic scoring and Task B on trait-specific scoring. It introduces a new (and first of its kind) dataset of 1,265 Arabic essays, annotated with *holistic* and *trait-specific* scores, including relevance, organization, vocabulary, style, development, mechanics, and grammar. The main goal of *TAQEEM* is to address the scarcity of standardized benchmarks and high-quality resources in Arabic AES. *TAQEEM* 2025 attracted 11 registered teams for Task A and 10 for Task B, with a total of 5 teams, across both tasks, submitting system runs for evaluation. This paper presents an overview of the task, outlines the approaches employed, and discusses the results of the participating teams.

1 Introduction

Automated Essay Scoring (AES) systems automatically assess the writing quality of essays, providing holistic scores, trait-specific (i.e., multi-dimensional) scores, or both. Effective AES systems have brought benefits, such as saving teachers time and effort, and producing less-biased and consistent results. This is crucial in large-scale assessments, such as international exams with thousands of participants, making AES a high-stakes application (Burstein, 2013).

There are two AES paradigms: *prompt-specific* and *cross-prompt*. The dominant *prompt-specific* AES trains and tests models on essays from the same prompt[1] (Taghipour and Ng, 2016). This setup achieves high performance, but requires a large amount of labeled data for the target prompt. In contrast, *cross-prompt* AES trains a model on a set of source prompts and tests it on unseen target prompts (Ridley et al., 2021). This approach is more practical, reducing the reliance on large labeled data for every new prompt. However, it faces challenges in achieving high performance due to source and target prompt variations.

Despite significant advances in AES for languages such as English (Klebanov and Madnani, 2022), Arabic AES remains understudied due to the lack of publicly annotated datasets for Arabic essay scoring, and the language's complex nature. Nevertheless, there has been some work on prompt-specific Arabic AES (Gaheen et al., 2021, 2020); however, to the best of our knowledge, no work has been done on cross-prompt Arabic AES. This motivated us to organize the first shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions (*TAQEEM*).[2] The task focuses on developing models for the automatic assessment of Arabic essays, both at a holistic level and across several traits. Through *TAQEEM*, we aim to advance research in Arabic AES by releasing the *first publicly available dataset* of 1,265 Arabic essays annotated with holistic and seven traits: relevance (الصلة بالموضوع), organization (الهيكل العام), vocabulary (المفردات), style (الأسلوب والتماسك البنائي), development (الأفكار والمضمون), mechanics (الإملاء والترقيم), and grammar (البناء والتراكيب).

TAQEEM 2025[3] focuses on cross-prompt AES setup, where models are evaluated on their ability to generalize to unseen prompts by leveraging knowledge learned from different labeled

[1] A prompt is the text of a specific essay writing task.

[2] Pronounced in Arabic as "تَقْيِيم".

[3] https://sites.google.com/view/taqeem-2025

Proceedings of The Third Arabic Natural Language Processing Conference, pages 966–976
November 8-9, 2025 ©2025 Association for Computational Linguistics

Team	Tasks	Team Size	Affiliations
912 (Vu and Đáng Văn, 2025)	A	2	University of Information Technology (UIT), VNUHCM, Vietnam
MarsadLab (Bessghaier et al., 2025)	A	3	University of Kairouan, Northwestern University, Hamad bin Khalifa University
ARxHYOKA (Alnajjar et al., 2025)	B	2	Nara Institution of Science and Technology, Tokyo University of Science
Taibah (Almarwani et al., 2025)	A,B	3	Taibah University
ANLPers3	A	5	Prince Sultan University

Table 1: Participating teams in *TAQEEM* 2025.

source prompts, thereby ensuring robustness and adaptability in real-world applications. *TAQEEM* 2025 includes two subtasks: (A) **Holistic Scoring**, which involves predicting a single overall score for a given essay reflecting its general quality, and (B) **Trait-specific Scoring**, which involves predicting separate scores for individual traits of the essay.

TAQEEM 2025 attracted registrations by 11 teams for Task A and 10 teams for Task B. However, in the final evaluation phase, only 4 teams submitted a total of 9 runs for Task A, while 2 teams contributed 4 runs for Task B. With one team actively involved in both tasks, this resulted in a total of 5 unique teams overall participating in *TAQEEM* 2025. Table 1 lists the participating teams, along with their affiliations and team sizes.

The remainder of the paper is organized as follows. Section 2 reviews related work on AES datasets and systems. Section 3 formally defines the two tasks, presents the dataset, and describes the evaluation setup. Section 4 discusses the approaches adopted by the participating teams along with their performance results. Finally, Section 5 concludes with final thoughts on future directions.

2 Related Work

This section reviews prior AES research, with particular emphasis on Arabic datasets and systems.

Datasets Progress in English AES has been driven by large public datasets such as ASAP[4] and ELLIPSE[5] with around 13,000 and 6,500 annotated essays, respectively. In contrast, Arabic AES lags behind due to the scarcity of annotated datasets, which are often small, limited in annotations, or not publicly accessible. For instance, the Zayed Arabic English Bilingual Undergraduate Corpus (ZAEBUC) (Habash and Palfreyman, 2022) contains 214 essays but lacks holistic and trait annotations. The Arabic Learner Corpus (ALC)[6] includes 1,585 essays, though its annotations are not publicly available. More recently, QAES dataset (Bashendy et al., 2024) was released with only 195 essays annotated with holistic and trait scores, building on the larger Qatari Corpus of Argumentative Writing (QCAW) (Ahmed et al., 2024).

Other datasets with holistic or trait annotations exist but are not public, such as Abbir (Alghamdi et al., 2014), which contains essays from Saudi university students with holistic scores from 1 to 6, and AAEE (Azmi et al., 2019), which evaluates essays based on semantic analysis, writing style, and spelling accuracy. Other datasets was collected for Arabic short-answer scoring (Abdeljaber, 2021; Ouahrani and Bennouar, 2020). Despite prior efforts, Arabic AES research still lacks a publicly available dataset that provides both essays and corresponding scores. Our shared task addresses this gap by releasing *TAQEEM* dataset, annotated with holistic and seven-trait scores, thereby making a substantial contribution to Arabic AES resources.

Systems Despite limited datasets, several studies have explored Arabic AES. Early work relied on traditional approaches that required extensive feature engineering (Alqahtani and Alsaif, 2020; Alsanie et al., 2022; Sayed et al., 2025). Other methods incorporated reference essays for scoring (Abdeljaber, 2021; Alobed et al., 2021a; Al Awaida et al., 2019; Alobed et al., 2021b). More recent efforts have advanced Arabic AES through the use of AraBERT and large language models (LLMs). Ghazawi and Simpson (2024) fine-tuned AraBERT with notable success, while Machhout and Zribi (2024)

[4] https://www.kaggle.com/c/asap-aes

[5] https://github.com/scrosseye/ELLIPSE-Corpus

[6] https://www.arabiclearnercorpus.com

Trait	Description
Relevance	Relevance of the essay to the prompt
Organization	The structure of the essay
Vocabulary	Precision and variety of word choice
Style	Linking words and transition phrases
Development	The support and clarity of ideas
Mechanics	Spelling and punctuation
Grammar	Accuracy of grammatical structures
Holistic	The overall quality score

Table 2: A brief description of the scoring traits.

improved its performance by integrating handcrafted features for relevance evaluation. Mahmoud et al. (2024) further optimized AraBERT using parameter-efficient tuning strategies. In parallel, Ghazawi and Simpson (2025) tested LLM-based approaches, experimenting with different LLMs under different prompting setups.

3 TAQEEEM 2025

In this section, we formally define *TAQEEM* 2025 subtasks, introduce the dataset, and elaborate on the evaluation setup.

3.1 Task Description

TAQEEM 2025 comprises two subtasks: Task A focuses on holistic scoring, while Task B targets trait-specific scoring.

Task A: Holistic Scoring The task is defined as follows: Given a set of source prompts P_{src}, the aim is to train a *holistic* scoring model using those prompts to score essays written for an unseen target prompt $p_{trg} \notin P_{src}$. The model should produce a single holistic score that reflects the overall writing quality of each essay.

A writing prompt p in this task is defined as a tuple (a_p, E_p), where a_p is the textual description of the writing task of the prompt and E_p is a set $\{(e, h_e)\}$ of essays written for the prompt p; each essay e is associated with a holistic score h_e.

Task B: Trait-specific Scoring The task is defined as follows: Given a set of source prompts P_{src}, the aim is to train a *trait-specific* scoring model using those prompts to score essays written for an unseen target prompt $p_{trg} \notin P_{src}$. For each essay written for p_{trg}, the model should produce a score *for each trait* that reflects the quality of the essay for that trait.

Data	Prompt	Type	Size	Len.
Training	1	Explanatory	215	137
Training	2	Persuasive	210	150
Test	9	Explanatory	420	153
Test	10	Persuasive	420	166

Table 3: *TAQEEM* 2025 dataset statistics. Size indicates number of essays, and length is indicated in words.

A writing prompt p in this task is defined as a tuple (a_p, T_p, E_p), where a_p is the textual description of the writing task of the prompt, T_p is a set $\{(t, r_t)\}$ of traits; each trait t is associated with a rubric r_t, and E_p is a set $\{(e, \{s_{e,t}\})\}$ of essays written for the prompt p; each essay e is associated with a score $s_{e,t}$ for each trait $t \in T_p$. While each prompt has its own trait rubrics, those rubrics are usually common across different prompts for specific traits.

In *TAQEEM* 2025, all essays of all prompts are annotated for the same seven traits: Relevance (REL), Organization (ORG), Vocabulary (VOC), Style (STY), Development (DEV), Mechanics (MEC), and Grammar (GRA). We note that the holistic (HOL) score, used in Task A, represents the sum of all trait scores. Table 2 provides a brief description of each trait.

3.2 Dataset

The absence of standardized Arabic essay corpora, even modestly sized ones, has slowed the progress in Arabic AES. To address this gap, we introduce a *novel* dataset[7] of 1,265 Arabic essays written by native high school and first-year university students under test-like conditions. The essays span 4 distinct writing prompts, ensuring diversity in content and structure. Table 3 provides an overview of the prompts used in both the training and test sets, including the number of essays per prompt and their average length in words. Notably, the test set is, unusually, larger than the training set. This is because, at the time of releasing the training data, only 425 fully annotated essays from prompts 1 and 2 were available for use. The remaining essays (from prompts 9 and 10) were still undergoing annotation, which was finalized by the time of the test set release, thereby allowing these additional essays to be included for evaluation.

[7] https://gitlab.com/bigirqu/taqeem2025

المعرّف	نص الموضوع و الموضوع	الدرجات
011069	**نص الموضوع:** باتَ اِهْتمام وحماس المراهقين لِتعلُمْ رِياضةٍ جديدة أو الاِنتظام في حُضور التَدْريبات الرياضية الخاصة بِها يَتضَاءل يومًا بعد يوم؛ حتَىْ صَارَ يدُق نَاقُوسَ خَطرٍ مقلق ينْذر بوجُود جيلٍ ضَعيفِ البنيَة (الجسم). اكتب مقالًا مكوَنا من ثلاثمائة (٣٠٠) كلمة توضّح فيه أسباب انتشار هذه الظاهرة، مُراعيًا سمات المقال التفسيري، وسلامة اللغة، ومُوظفًا علامات الترقيم وأدوات الربط بشكلٍ صحيح. **الموضوع:** ان الرياضه في يومنا الحالي بات الاهتمام بها من قبل المراهقين لتعلم رياضه جديده او في الحضور للتدريبات الخاصه، ويوم بعد يوم يقل النشاط ةيزداد الكسل لديهم، وهذا خطر على صحتهم من قبل تاكيد الاطباء عليه. وكما ان زاد قل نشاط المراهقين في وقتنا الحالي يجب علينا ان نعالج هذا قبل ان يشكل خطرا اكثر عليهم ويكونون جيل ضعيف البنيه. ومن اسباب هذا الكسل عدم اهتمام وتفرغ بعض اولياء الامور لاطفالهم وتشجيعهم على الرياضه، وكذلك كثرة جلوس الاطفال على الاجهزه الالكترونيه مما يسبب الكسل في الحركه والخمول لدى الطفل المراهق، وايضا في بعض المدارس عدم ممارسة الرياضه في بداية اليوم الدراسي و عدم التوعيه بأهميه الرياضه في حياتنا من الناحيه الصحيه وكم هي تشكل خطرا على الجسم اذا ما مارسوا الرياضه، وكذلك عدم تعليم الطالب وتشجيعه من قبل المدرسه لتعلم رياضه جديد وعمل انشطه وفعاليات بين فترات للطلبه المراهقين. وجملة القول، يجب على الاسره الاهتمام باطفالهم والتفرغ لهم وكذلك المدرسه في عمل لهم نشاطات والتعرف على الرياضات الجديده لتعلمها.	الصلة بالموضوع: ٢ الهيكل العام: ٤ المفردات: ٣ الأسلوب والتماسك: ٣ الأفكار و المضمون: ٣ الإملاء و الترقيم : ٢ البناء و التراكيب: ٣ المجموع الكلي: ٢٠
100099	**نص الموضوع:** يرى البعض أنَ تقليل عبء الواجبات المنزليَةْ على الطَلْبة لا يؤثِرْ على أدائِهم الأكاديميّ، ويرفع رفاهيّتِهم، ويزيد الأوقات التي يقضونها مع أسرهم. كيف ترى قضية تقليل الواجبات المنزلية؟ اكتب مقالًا مكونًا من ثلاثمائة (٣٠٠) كلمة لتقنع القارئ بوجهة نظرك في هذا الموضوع موظّفًا الأدلة والحجج الدّاعمة لهذا الرأي، ومراعيًا أساليب الإقناع، ومستخدمًا علامات الترقيم وأدوات الربط المناسبة. **الموضوع:** إنه مما لا شك فيه أن الواجبات المدرسية والتقييمات المنزلية أصبحت وسيلة أساسية يرتكز عليها المدرسين في تثبيت المعلومة لدى الطلبة. يُعطى الطالب الواجب المدرسي ويتم تخصيص مكافأة لمن أنجز أو عقوبة لمن لم ينجز العمل الموكّل به. وبذلك، يضمن المعلم والكادر التعليمي أن الطالب قد خصص وقتًا للدراسة، وأنه حاول ليرى نقاط الضعف لديه في هذا الدرس. لكن يرى البعض أن الواجبات المنزلية قد تزيد من العبء على الطالب و تحمله ضغوطًا أخرى قد يكون بغنى عنها . إن تكليف الطالب بأداء الواجبات المدرسية والحرص على تقديمها في الموعد المناسب يرفع من حس المسؤولية لدى الطالب : وذلك بتقليل الاعتماد على الغير ومحاولة الاعتماد على النفس وضبط أهدافه وتنظيم وقته ليناسب وقت تسليمه للواجب . إن الواجبات والتكليفات أساسًا من جميع النواحي العملية ترفع من حس المسؤولية عامةً ، حتى في ديننا الحنيف قد أمرنا الله تعالى بأداء خمسة فروض وواجبات مكلفة في كل يوم ، يعاقب من لم يؤدها ويجازى من قام بها . من هذا المنطلق ومن هذه الفكرة أقول بأن الواجبات والفروض والتكليفات التي تفرض على الطالب إنما لزيادة حرصه على دراسته وعدم التراخي وإشعال حس اليقظة والمسؤولية تجاه النفس . أضف إلى ذلك، أن أداء الطالب لواجباته المدرسية يرفع من المستوى الأكاديمي للطالب ويثبت المعلومات في ذاكرته: وذلك لأن الطالب قد يتذكر المعلومة التي التقطها من المعلم في الفصل لفترة قصيرة فقط ثم ينساها في حال أنه لم يراجعها ويثبتها؛ فعلميًا الذاكرة قصيرة المدى تخزن المعلومة الحسية في حدود ساعتين إلى ثلاث ساعتين ثم تبدأ بالتلاشي مع الوقت، أما إذا تم تكرار تلك المعلومة فإنها ستثبت مدة أطول، وكلما زاد التكرار زادت المعلومة ثباتًا واستقرارًا وزادت معها الفائدة العائدة منها. ختامًا أقول: إن تكليف الطالب بأداء مهامه الدراسية لهو من مصلحته الشخصية أولاً وقبل كل شيء، و قد يرى الطالب بأنه زيادة تكليف وأنه عبء عليه، لكن في المقابل يجب توعية الطلبة بضرورة المتابعة والدراسة حتى وإن لم يكلفوا بواجبات مدرسية؛ لتكون الفائدة خالصة ولينمو لديهم حس المسؤولية، وقيم ضبط ومجاهدة النفس، والصبر.	الصلة بالموضوع: ٢ الهيكل العام: ٥ المفردات: ٥ الأسلوب والتماسك: ٥ الأفكار و المضمون: ٥ الإملاء و الترقيم : ٥ البناء و التراكيب: ٤ المجموع الكلي: ٣١

Table 4: Annotated Essays from *TAQEEM* 2025 dataset.

Annotation Process The annotations were conducted by two main native Arabic language specialists, with a third annotator resolving disagreements. Annotators were selected for their experience in teaching and assessing Arabic writing. To ensure score reliability, all annotators received training sessions to understand the assessment rubric and maintain consistent annotation procedures. The rubric itself was adapted from the Core Academic Skills Test (CAST) developed by the Qatar University Testing Center (QUTC).[8] A full detailed English-translated version of the rubric is in Appendix A. Each essay was annotated across 7 traits: REL, ORG, VOC, STY, DEV, MEC, and GRA, along with an overall quality score (HOL) computed as the sum of all trait scores. Traits are rated on a 0 to 5 scale, except for REL, which is from 0 to 2, and the HOL, which is from 0 to 32, all using 1-point increments. Table 4 shows two essays from the *TAQEEM* 2025 dataset, a training essay (ID 011069) from an explanatory prompt (Prompt 1) and a test essay (ID 100099) from a persuasive prompt (Prompt 10), along with their prompts and scores.

Inter-Annotator Agreement We assessed annotation quality using the Quadratic Weighted

[8] https://www.qu.edu.qa/sites/en_US/testing-center/TestDevelopment/cast

Kappa (QWK) (Cohen, 1968), averaging trait-level scores per prompt to obtain prompt-level agreement. The resulting average agreements were 0.692 for Prompt 1, 0.640 for Prompt 2, 0.525 for Prompt 9, and 0.676 for Prompt 10. According to the scale outlined by Landis and Koch (1977), Prompts 1, 2, and 10 fall within the range of *substantial* agreement, while Prompt 9 shows *moderate* agreement, possibly due to less precise wording that made it more open to interpretation and increased variability in annotators' judgments. Nevertheless, the overall results indicate strong rater consistency across prompts.

This dataset is used in both subtasks of *TAQEEM* 2025, with distinct evaluation targets. Task A (Holistic Scoring) uses the holistic score assigned to each essay, whereas Task B (Trait-specific Scoring) uses the seven individual trait scores. Although this dataset is limited in scale, it represents a carefully curated first step resource to address data scarcity in Arabic AES.

3.3 Evaluation Setup

This section outlines the setup used to evaluate participating systems in *TAQEEM* 2025. We describe the leaderboard and repository infrastructure provided to participants, as well as the evaluation measures adopted to ensure consistent, and reproducible comparisons across submitted systems.

3.3.1 Leaderboard and Repository

The leaderboard for both Task A[9] and Task B[10] was hosted on Codabench, providing participants a platform to submit their runs, evaluate system outputs, and benchmark performance. Each team was required to submit their predictions in a single file, referred to as a run file. Submissions were restricted to a maximum of 30 runs on the development set and up to 3 runs on the test set. Typically, each run represented a distinct system or model.

To facilitate the submission process, we made the submission checker and evaluation scripts available through the shared task repository. These resources enabled participants to validate their runs before leaderboard submission. Additionally, we released a regression-based baseline by fine-tuning AraBERTv02 (Antoun et al.), along with the corresponding code, in the same repository.

Team	Run	QWK	MSE	RMSE
Taibah	1	**0.751**	**25.44**	**5.01**
912	1	0.673	28.51	5.33
912	2	0.673	28.51	5.33
ANLPers3	1	0.650	31.68	5.62
ANLPers3	2	0.642	28.28	5.28
baseline	0001	0.639	29.01	5.37
ANLPers3	3	0.602	29.73	5.45
Taibah	2	0.488	33.15	5.73
MarsadLab	1	0.438	50.56	7.07
MarsadLab	2	0.438	50.56	7.07

Table 5: Task A performance results on the test set. Bold values are the best for each measure.

3.3.2 Evaluation Measures

The primary evaluation metric for *TAQEEM* 2025 is the Quadratic Weighted Kappa, a standard AES performance metric that quantifies the agreement between human-assigned scores and system predictions. Additionally, we report the mean squared error (MSE) and the root mean squared error (RMSE) to provide a more comprehensive analysis of model performance, as these metrics capture the magnitude of prediction errors, penalize larger deviations more heavily, and allow for direct comparison of error scales across models.

The subtasks are evaluated independently. Task A is assessed based on the average QWK of the holistic score across the test prompts. For Task B, the average QWK for each trait across the test prompts is measured separately, and teams are ranked based on the average QWK over all traits.

4 Participating Systems and Results

This section presents the participating systems and their performance in *TAQEEM* 2025, highlighting the methods used and the corresponding evaluation results for both subtasks.

4.1 Task A: Holistic Scoring

Task A attracted 4 teams in total, each adopting distinct methodological approaches, resulting in 9 runs submitted on the test set. The top-ranked team, Taibah) (Almarwani et al., 2025), employed a rubric-guided few-shot prompting strategy based on GPT-4o, utilizing exemplars to assess the holistic quality of essays. The 912 team (Vu and Đặng

[9] https://www.codabench.org/competitions/9282/
[10] https://www.codabench.org/competitions/9295/

Team	Run	QWK								MSE	RMSE
		REL	ORG	VOC	STY	DEV	MEC	GRA	Avg.		
Taibah	1	0.562	0.668	0.642	**0.678**	**0.703**	**0.644**	**0.664**	**0.652**	**0.762**	**0.857**
ARxHYOKA	1	0.553	0.709	0.633	0.654	0.640	0.515	0.580	0.612	0.760	0.848
ARxHYOKA	2	**0.585**	0.711	0.646	0.666	0.647	0.477	0.544	0.610	0.758	0.845
ARxHYOKA	3	0.545	**0.712**	**0.653**	0.620	0.629	0.482	0.506	0.592	0.797	0.867
Baseline	-	0.155	0.591	0.574	0.572	0.458	0.445	0.513	0.472	1.005	0.990

Table 6: Task B performance results on the test set. The best score for each metric is highlighted in bold.

Văn, 2025) adopted a pre-trained Arabic encoder (AraBERTv02) with a lightweight single-layer MLP head, coupled with a distribution-sensitive weighted MSE loss to address score imbalance. The MarsadLab system (Bessghaier et al., 2025) was also built on a fine-tuned AraBERT model, integrating lexical features into the embeddings to predict essay scores.

In terms of performance, which is summarized in Table 5, teams were ranked by their highest average QWK score across all test prompts. The highest performing system, submitted by the Taibah team, achieved a QWK of 0.751, significantly outperforming the baseline of the shared-task (QWK of 0.639). Team 912 followed by two identical runs reaching a QWK of 0.673. Team ANLPers3 also delivered competitive systems, with their best run achieving a QWK of 0.650. The two runs of MarsadLab resulted in the lowest performance across submissions (QWK of 0.438), and it was the only team that did not outperform the baseline. Overall, three of the four teams submitted at least one run that outperformed the baseline, reflecting both the effectiveness and diversity of the applied approaches.

4.2 Task B: Trait-specific Scoring

Task B featured two participating teams, who together submitted four runs on the test set, implementing different approaches for trait-specific scoring. Notably, the Taibah team, which had also ranked first in Task A, once again secured the top position in Task B. They adopted a GPT-4o-based few-shot prompting approach, leveraging trait-specific rubrics to achieve fine-grained scoring (Almarwani et al., 2025). The second-ranked team (ARxHYOKA) (Alnajjar et al., 2025) explored a broader methodological spectrum, including GPT-based few-shot prompting, fine-tuned BERT-based models, classical machine learning approaches with embeddings and handcrafted features, and fine-tuned text-generation LLMs. Their best-performing configuration used GPT-4.1 with 10-shot chain-of-thought prompting.

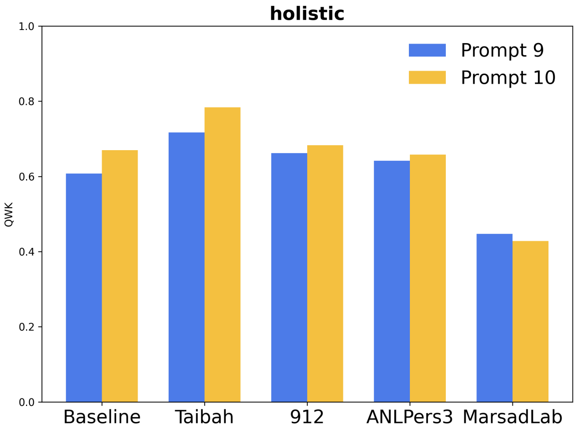


Figure 1: Performance of Task A teams across test prompts. The best submitted run was considered.

Performance was evaluated based on the average QWK across all 7 traits. The top-performing run, submitted by Taibah, achieved an average QWK of 0.652, with MSE of 0.762 and RMSE of 0.857, substantially outperforming the shared-task baseline (average QWK of 0.472, MSE of 1.005, RMSE of 0.990). ARxHYOKA also outperformed the baseline, reaching an average QWK of 0.612. These results underscore the potential of prompting strategies for trait-specific scoring in Arabic AES. Table 6 presents the test results for Task B, reporting QWK, MSE, and RMSE measures.

4.3 Analysis and Discussion

This section provides a detailed analysis of the results from two perspectives: *trait-level* performance and *prompt-level* performance. Figure 1 shows Task A performance for the holistic scoring,

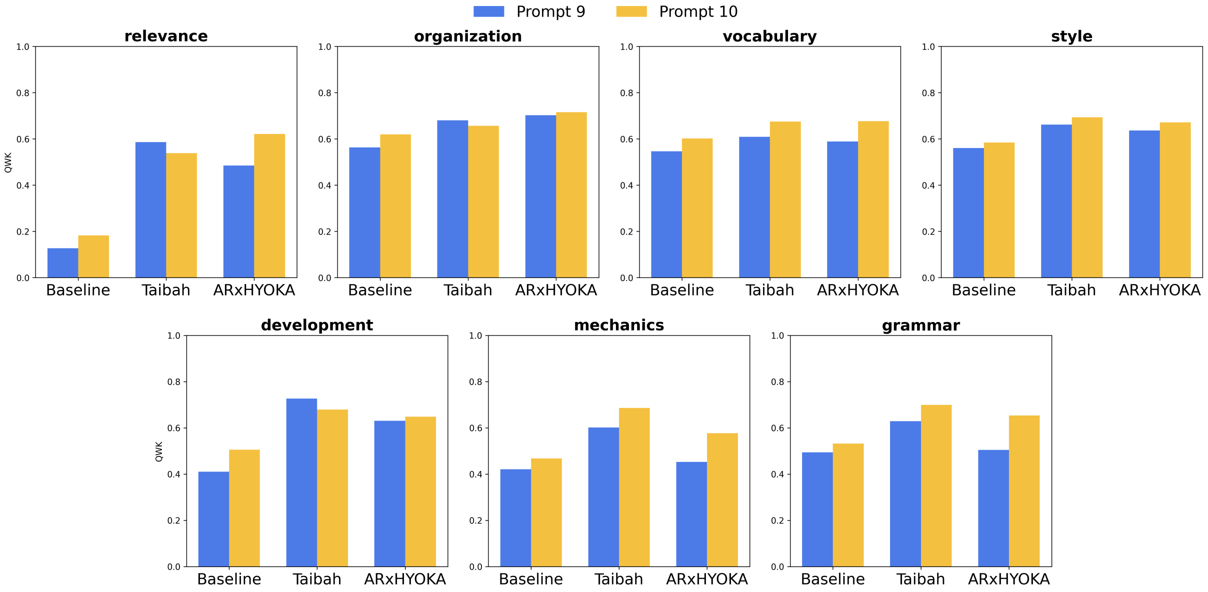


Figure 2: Trait-level performance of Task B teams on prompt 9 (Explanatory) and prompt 10 (Persuasive). The best submitted run was considered.

whereas Figure 2 illustrates Task B performance for the trait-specific scoring. In both cases, the figures report results from each team's best submitted run, providing a clear view of top-performing approaches for each task across the two test prompts.

Trait-level Analysis Figure 2 reveals clear differences in teams performance across the different traits. Notably, the REL trait appears to be the most challenging, as evidenced by the highest QWK score being only 0.585, achieved by the ARxHYOKA team across the two test prompts. The baseline, which relies on fine-tuning AraBERTv2, struggled the most with the REL trait. This suggests that smaller encoders like AraBERT have difficulty capturing the semantic alignment between essays and prompts. In contrast, both Taibah and ARxHYOKA show substantial improvements in REL, demonstrating that leveraging the advanced capabilities of GPT-4o and GPT-4.1 through few-shot prompting significantly enhances performance on this semantically complex trait. The MEC trait also proved difficulty, with an average QWK of 0.580 across the two participating teams and test prompts, reflecting the difficulty of capturing fine-grained linguistic correctness, such as punctuation, spelling, and syntax. Traits VOC and GRA showed moderate performance across teams, while ORG, STY, and DEV exhibited comparatively higher and more consistent performance.

Prompt-level Analysis Performance also varies depending on the prompt type. From Figures 1 and 2, it is evident that Prompt 10 (Persuasive) generally resulted in higher performance across most traits and teams compared to Prompt 9 (Explanatory). This pattern suggests that persuasive writing, which typically follows a predictable and structured format (e.g., a clear thesis statement, supporting arguments, counterarguments, and a conclusion), is easier for models to capture. Explanatory essays, on the other hand, exhibit greater structural and stylistic diversity, making it more difficult for models to identify consistent patterns.

Overall, these results clearly indicate that GPT-4-based models (Taibah & ARxHYOKA) generally outperform fine-tuned BERT models (Baseline, 912, MarsadLab) across most traits. This shows the potential of LLMs for automated essay scoring. Their ability to understand complex language and capture nuanced relationships leads to significantly higher agreement with human scores. While fine-tuned BERT models provide a reasonable baseline, they struggle to match the performance of LLMs.

5 Conclusion

Automated Essay Scoring has seen notable progress in writing evaluation, yet the development of AES systems tailored for the Arabic language remains very limited. This scarcity motivated the organization of TAQEEM, the first shared task dedicated to Arabic AES, aiming to foster state-of-the-art research in this area, with a novel dataset of 1,265 essays across four different writing prompts.

TAQEEM 2025 attracted 15 researchers and practitioners across five teams from different institutions. It comprised two cross-prompt subtasks: holistic scoring (Task A), with four participating teams, and trait scoring (Task B), with two teams. The participating teams explored diverse solutions, including fine-tuning transformer-based models and employing classical machine learning approaches. However, as expected, LLMs were heavily adopted by multiple teams, achieving state-of-the-art performance and outperforming the baseline. For task A, the teams employed different solutions that mainly focused on fine-tuning different transformer-based models and prompting LLMs using different prompting techniques. For task B, the best results were achieved with few-shot in-context learning and chain-of-thought prompting using GPT-4 variants.

Overall, *TAQEEM* 2025 established the first benchmark for Arabic AES, providing a foundation for future research and community efforts to develop AES systems for the Arabic language. In the next iteration, we plan to expand the shared task by incorporating a larger training set that encompasses a wider range of essay types, topics, and student populations, thereby fostering deeper research advancements and broader community contributions in this area.

6 Limitations

One key limitation of TAQEEM is the size and diversity of the dataset. Although it provided a useful benchmark for Arabic AES, the training and test sets were relatively small and may not fully capture the variety of essay topics, writing styles, or proficiency levels. Moreover, the test set was larger than the training set due to the challenges and time required to provide high-quality annotated data. This limitation could affect the generalizability of the models trained and evaluated in this shared task. Another limitation is the small number of participating teams, which reduces the variety of approaches evaluated.

Acknowledgments

We would like to thank our dedicated annotators who contributed to building the dataset. We also extend our sincere thanks to the Qatar University Testing Center (QUTC), the Ministry of Education, the participating schools, and the students for their essential role in making this work possible. This work was supported by NPRP grant# NPRP14S-0402-210127 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Hikmat A. Abdeljaber. 2021. Automatic arabic short answers scoring using longest common subsequence and arabic wordnet. *IEEE Access*, 9:76433–76445.

Abdelhamid M. Ahmed, Xiao Zhang, Lameya M. Rezk, and Wajdi Zaghouani. 2024. Building an Annotated L1 Arabic/L2 English Bilingual Writer Corpus: The Qatari Corpus of Argumentative Writing (QCAW). *Corpus-based Studies across Humanities*, 1(1):183–215.

Saeda A Al Awaida, Bassam Al-Shargabi, and Thamer Al-Rousan. 2019. Automated arabic essay grading system based on f-score and arabic worldnet. *Jordanian Journal of Computers and Information Technology*, 5(3).

Mansour Alghamdi, Mohamed Alkanhal, Mohamed Al-Badrashiny, Abdulaziz Al-Qabbany, Ali Areshey, and Abdulaziz Alharbi. 2014. A hybrid automatic scoring system for Arabic essays. *AI Communications*, 27(2):103–111.

Nada Almarwani, Alaa Alharbi, and Samah Aloufi. 2025. Taibah at TAQEEM 2025: Leveraging GPT-4o for Arabic Essay Scoring. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

Mohamad Alnajjar, Ahmad Almoustafa, Tomohiro Nishiyama, Shoko Wakamiya, Eiji Aramaki, and Takuya Matsuzaki. 2025. ARxHYOKA at TAQEEM2025: Comparative Approaches to Arabic Essay Trait Scoring. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

Mohammad Alobed, Abdallah M M Altrad, and Zainab Binti Abu Bakar. 2021a. A comparative analysis of euclidean, jaccard and cosine similarity measure and arabic wordnet for automated arabic essay scoring. In *2021 Fifth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pages 70–74.

Mohammad Alobed, Abdallah MM Altrad, Zainab Binti Abu Bakar, and Norshuhani Zamin. 2021b. Automated arabic essay scoring based on hybrid stemming with wordnet. *Malaysian Journal of Computer Science*, pages 55–67.

Abeer Alqahtani and Amal Alsaif. 2020. Automated Arabic essay evaluation. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 181–190, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Waleed Alsanie, Mohamed I Alkanhal, Mohammed Alhamadi, and Abdulaziz O Alqabbany. 2022. Automatic scoring of arabic essays over three linguistic levels. *Progress in Artificial Intelligence*, pages 1–13.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Aqil M. Azmi, Maram F. Al-Jouie, and Muhammad Hussain. 2019. Aaee – automated evaluation of students' essays in arabic language. *Information Processing Management*, 56(5):1736–1752.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Eman Zahran, Hamdo Elhuseyin, Tamer Elsayed, Walid Massoud, and Houda Bouamor. 2024. QAES: First publicly-available trait-specific annotations for automated scoring of Arabic essays. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 337–351, Bangkok, Thailand. Association for Computational Linguistics.

Mabrouka Bessghaier, Md. Rafiul Biswas, Amira Dhouib, and Wajdi Zaghouani. 2025. Marsadlab at TAQEEM 2025: Prompt-Aware Lexicon-Enhanced Transformer for Arabic Automated Essay Scoring. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

Jill Burstein. 2013. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Marwa M. Gaheen, Rania M. ElEraky, and Ahmed A. Ewees. 2020. Optimized neural network-based improved multiverse optimizer algorithm for automated arabic essay scoring. *International Journal of Scientific & Technology Research*, 9:238–243.

Marwa M Gaheen, Rania M ElEraky, and Ahmed A Ewees. 2021. Automated students arabic essay scoring using trained neural network by e-jaya optimization to support personalized system of instruction. *Education and Information Technologies*, 26:1165–1181.

Rayed Ghazawi and Edwin Simpson. 2024. Automated essay scoring in arabic: a dataset and analysis of a bert-based system. *arXiv preprint arXiv:2407.11212*.

Rayed Ghazawi and Edwin Simpson. 2025. How well can llms grade essays in arabic? *arXiv preprint arXiv:2501.16516*.

Nizar Habash and David Palfreyman. 2022. ZAEBUC: An annotated Arabic-English bilingual writer corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 79–88, Marseille, France. European Language Resources Association.

Beata Beigman Klebanov and Nitin Madnani. 2022. *Automated essay scoring*. Springer Nature.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Rim Aroua Machhout and Chiraz Ben Othmane Zribi. 2024. Enhanced bert approach to score arabic essay's relevance to the prompt. *Communications of the IBIMA*, 2024.

Somaia Mahmoud, Emad Nabil, and Marwan Torki. 2024. Automatic scoring of arabic essays: A parameter-efficient approach for grammatical assessment. *IEEE Access*.

Leila Ouahrani and Djamal Bennouar. 2020. AR-ASAG an ARabic dataset for automatic short answer grading evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2634–2643, Marseille, France. European Language Resources Association.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13745–13753.

Marwan Sayed, Sohaila Eltanbouly, May Bashendy, and Tamer Elsayed. 2025. Feature engineering is not dead: A step towards state of the art for arabic automated essay scoring. In *Proceedings of the Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Trong-Tai Dam Vu and Thìn Đáng Văn. 2025. 912 at TAQEEM 2025: A Distribution-aware Approach to Arabic Essay Scoring. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

A Grading Rubric

For annotating *TAQEEM*2025 dataset, we utilized the rubric from the Core Academic Skills Test

(CAST) designed by the Qatar University Testing Center (QUTC)[11],which is provided in Arabic. This rubric guided the scoring of seven traits: relevance (REL), organization (ORG), vocabulary (VOC), style (STY), development (DEV), mechanics (MEC), and grammar (GRA). An English-translated version of the CAST grading rubric for each trait is provided in Table 7.

[11] https://www.qu.edu.qa/sites/en_US/testing-center/TestDevelopment/cast

Trait	**1**	**2**	**3**	**4**	**5**
REL	Partially relevant to the topic	Completely relevant to the topic			
ORG	The introduction and conclusion are absent. There is no organization or sequence between paragraphs.	Either the introduction or conclusion is absent. There is no organization or sequence between paragraphs.	The text is well-organized and contains an introduction and conclusion, but the body has one paragraph (or two paragraphs) that lacks good coherence.	The text is well-organized, contains an appropriate introduction and conclusion, and has two to three body paragraphs that are sequential and coherent.	The text is well-organized and contains an introduction that introduces the topic, a conclusion that effectively concludes the text, and two to three body paragraphs that are sequential and well-connected.
VOC	Use of a limited range of vocabulary and phrases that do not make sense together, with repetition and lexical errors, and generally inappropriate vocabulary that obscures meaning.	Use of a basic range of vocabulary, with repetition, lexical errors, and many inappropriate choices that may obscure meaning.	Use a sufficient range of vocabulary, with some repetition and lexical errors, with a small number of inappropriate vocabulary that may obscure meaning.	Use of a good and appropriate range of vocabulary with few lexical errors, inappropriate choices without affecting meaning, and occasional use of idiomatic expressions.	Use of a broad, correct, and appropriate range of vocabulary with few occasional errors, showing good knowledge of idiomatic expressions and awareness of implicit levels of meaning.
STY	The text employs very basic linear connecting words such as "and" and "then."	Discourse develops as a simple list of points using only the most common connections.	Discourse develops directly as a linear sequence of points using common structural cohesion devices.	Discourse is clearly developed with main points supported by relevant details, appropriate use of different organizational patterns, and a range of structural cohesion devices.	Discourse is well developed, with good inclusion of subtopics and details and a good conclusion, always appropriate use of a variety of organizational patterns, and a wide range of structural cohesion devices.
DEV	Content is not related to the subject; ideas are random and lack coherence, sequence, and evidence.	Content is somewhat related; ideas are sequential but main idea disappears during writing, limited coverage, and poor use of supporting structures.	Content is completely related; ideas mostly follow sequence, main idea gradually disappears, some evidence present but disorganized.	Content is completely related; ideas are clear, organized, coherent, with main idea connected to sub-ideas, specific position adopted, some arguments and evidence presented coherently.	Content is completely related; ideas are clear, organized, coherent, main idea connected to sub-ideas, specific position adopted, arguments and evidence presented coherently, comprehensive coverage of opinions, and use of various persuasive methods.
MEC	Limited application of spelling rules.	Frequent spelling and punctuation errors.	Effectively applies standard formatting, paragraphing, spelling, and punctuation most of the time.	Effectively applies standard formatting, paragraphing, spelling, and punctuation with few errors.	Completely accurate paragraph organization, punctuation, and spelling, except for a few occasional pen slips.
GRA	Use a limited set of simple grammatical structures and sentence patterns with little flexibility or precision.	Correct use of some simple structures with frequent systematic errors that may obscure meaning.	Use a variety of grammatical structures, with notable errors that can sometimes obscure meaning.	Good use of variety of structures with rare errors and minor imperfections that do not affect meaning.	Always correct and flexible use of a wide variety of grammatical constructions with occasional minor slips.
Note: A score of zero is given if the response is completely memorized or copied from the prompt, if the student did not attempt the task, or if the content is irrelevant to the given topic.					

Table 7: CAST Persuasive/Argumentative Writing Rubric - English Translation (Bashendy et al., 2024).

ARxHYOKA at TAQEEM2025: Comparative Approaches to Arabic Essay Trait Scoring

Mohamad Alnajjar
Nara Institute of Science and Technology
alnajjar.mohamad.al8@naist.ac.jp

Ahmad Almoustafa
Tokyo University of Science
1425501@ed.tus.ac.jp

Tomohiro Nishiyama
Nara Institute of Science and Technology
nishiyama.tomohiro.ns5@is.naist.jp

Shoko Wakamiya
Nara Institute of Science and Technology
wakamiya@is.naist.jp

Eiji Aramaki
Nara Institute of Science and Technology
aramaki@is.naist.jp

Takuya Matsuzaki
Tokyo University of Science
matuzaki@rs.tus.ac.jp

Abstract

Arabic automated essay scoring (AES) presents unique challenges due to the linguistic complexity of Arabic and the need for rubric-specific evaluation. In this paper, we present ARxHYOKA, our submission to TAQEEM2025 Task B, which targets trait-specific AES using the Core Academic Skills Test (CAST) rubric. We evaluate four approaches: (1) GPT-based few-shot prompting, (2) fine-tuning BERT-based models, (3) classical machine learning approaches with embeddings and handcrafted features, and (4) fine-tuning text-generation large language models (LLMs). Our best-performing system, GPT-4.1 with 10-shot CoT prompting, achieved the highest official score, outperforming all other approaches in average Quadratic Weighted Kappa (QWK) in the test phase. Fine-tuned BERT-based models performed on par with both the shared-task baseline and our GPT prompting setup in the development phase, while classical machine learning methods trailed these systems, and the fine-tuned Arabic LLM ranked last. We provide comparative analyses across systems to inform future research on Arabic AES.

1 Introduction

The TAQEEM2025 Task B (Bashendy et al., 2025) targets automated scoring of Arabic essays, evaluating seven traits defined by the Core Academic Skills Test (CAST) rubric.[1] A central challenge is cross-prompt generalization: systems trained on one prompt must accurately score essays from a different, unseen prompt. This task advances robust, rubric-aligned Arabic NLP evaluation and enables fair, scalable, and transparent assessment of student writing in high-stakes settings across real-world educational contexts. In our submission, we compared three main approaches: prompting, fine-tuning, and training traditional machine learning (ML) models. Our key findings are as follows:

- **GPT-based few-shot prompting** achieved the highest average QWK, outperforming the baseline in the test phase and closely matching it in the development phase. Performance was sensitive to the number and quality of examples as well as the language used in the prompt.
- **Fine-tuning BERT-based models** produced strong results close to the baseline in the development phase. Both Arabic-specific and multilingual models performed well.
- **Fine-tuning text-generation model Saka 14B** yielded poor results, suggesting that relatively small LLMs may not be optimal for this scoring task without further adaptation.
- **Classical ML approaches** remained competitive, with performance improving when linguistic features were combined with embeddings.

Code and prompts are available at our repository.[2]

2 Background

The task involves predicting numeric scores for seven traits: **Relevance** (0–2), **Organization**, **Vocabulary**, **Style**, **Development**, **Mechanics**, and **Grammar** (0–5 each). Essays are written in response to prompts that are either *explanatory* or *persuasive*, mimicking real classroom writing tasks.

The official dataset for TAQEEM2025 Task B is summarized in Table 1. It contains two prompt types in the training phase and two in the test phase, with essays of approximately 300 words each. All essays have been scored by expert raters using the official CAST rubrics for each trait.

[1] https://www.qu.edu.qa/en-us/testing-center/TestDevelopment/Pages/cast.aspx

[2] https://github.com/Mohamad-Alnajjar/ARxHYOKA

Proceedings of The Third Arabic Natural Language Processing Conference, pages 977–982
November 8-9, 2025 ©2025 Association for Computational Linguistics

Split	Prompt ID	Type	# Essays
Development Phase	1	Explanatory	215
Development Phase	2	Persuasive	210
Testing Phase	9	Explanatory	420
Testing Phase	10	Persuasive	420

Table 1: Dataset composition for TAQEEM2025 Task B.

This setup poses challenges for both linguistic coverage and cross-prompt adaptability, particularly for traits such as **Relevance**, where alignment with the prompt topic is critical.

3 System Overview

We present the systems explored for Arabic essay trait scoring, covering GPT-based prompting, fine-tuned BERT models, classical ML baselines, and a fine-tuned generative LLM.

3.1 GPT-Based Few-Shot Prompting

This system leverages GPT-4.1 to score essays based on in-context learning. The model relies entirely on the design of the prompt and the quality of examples provided. We tested prompts in both Arabic and English with different random sets of examples from the dataset in the development phase. The prompt includes:

- Detailed instructions for scoring.
- The CAST rubric.
- The essay type (explanatory or persuasive).
- The original writing prompt given to students.
- Instructions for structured output formatting.

We systematically compared model performance across:

- Arabic vs. English rubrics and prompts (translated using GPT-4.1).
- Number of in-context examples (0, 1, 5, and 10 shots).

3.2 Fine-Tuning BERT Models

We fine-tuned three encoder-only transformers: mDeBERTa-v3-base (He et al., 2021), XLM-R-large (Conneau et al., 2019), and CAMeLBERT-mix (Inoue et al., 2021), as independent systems (no ensembling). Essays are tokenized with each model's native tokenizer, truncated to 512 tokens, and passed to a 7-dimensional regression head to jointly predict the seven trait scores; at inference, continuous outputs are rounded and clamped to valid per-trait ranges.

3.3 Classical ML Approaches

We generated embeddings for each essay using `CAMeL-Lab/bert-base-arabic-camelbert-mix` (Inoue et al., 2021) and fed them to regression models to predict scores across seven traits. Inspired by (Bashendy et al., 2024), we also extracted 14 handcrafted linguistic features (listed in Table 10 in the Appendix) and evaluated the best-performing models during our experiments with and without these features.

We tested several pooling strategies and trained five regressors: LASSO, ElasticNet, Ridge, XGBoost, and Random Forest. Pooling strategies evaluated include:

1. `[CLS]` token
2. Average pooling
3. Average pooling + `[CLS]` token

3.4 Fine-Tuning Text-Generation LLM

We adapted `Sakalti/Saka-14B` (Sakalti, 2024), an open-source Arabic LLM, for trait-specific scoring using parameter-efficient fine-tuning (PEFT) via LoRA (Hu et al., 2021). To encourage rubric-grounded reasoning, we manually created two datasets:

- **Simple CoT**: 5–6 concise reasoning steps per trait focusing on essential rubric criteria.
- **Advanced CoT**: 7–8 detailed reasoning steps with deeper justification aligned to rubric criteria.

Each training instance concatenated the writing prompt, student essay, and trait-specific reasoning sequence with the gold score, encouraging the model to emulate human evaluation.

4 Experimental Setup

All experiments, including hyperparameter tuning and prompt engineering, used a cross-prompt setting: models were trained on **Explanatory** essays and tested on **Persuasive** essays. After selecting the best configurations, we retrained on the *union* of both essay types for the final submission.

- **Classical ML Approaches:** We performed 3-fold cross-validation using `scikit-learn`, optimizing for QWK. Initial experiments (Table 2) showed Ridge (AVG pooling, 0.521) and ElasticNet (CLS+AVG pooling, 0.527) as the strongest models, so we selected them for further evaluation. Incorporating handcrafted linguistic features (LF) improved results across both models (Table 3), with Ridge (AVG + LF) achieving the best QWK of 0.539. These findings highlight the complementary value of shallow linguistic cues when combined with transformer embeddings.

Model	CLS	AVG	CLS + AVG
Lasso	0.480	0.517	0.482
ElasticNet	0.474	0.518	0.527
XGBoost	0.472	0.495	0.479
Ridge	0.454	0.521	0.471
RandomForest	0.447	0.492	0.494

Table 2: Performance of different regression models using CLS, AVG, and CLS+AVG embeddings during experiments.

Pooling	Features	Ridge	ElasticNet
AVG	+ LF	0.539	0.529
AVG	– LF	0.524	0.514
AVG + CLS	+ LF	0.533	0.532
AVG + CLS	– LF	0.511	0.539

Table 3: Performance comparison of Ridge and ElasticNet across pooling strategies with and without linguistic features (LF) on the development dataset.

- **GPT-Based Few-Shot Prompting:** A structured CoT prompt was employed to score the essays. We used an English version of both the CAST rubric and the essay prompts, achieving better performance after translation (QWK improved from 0.539 to 0.579 compared to Arabic). We also tested different numbers of shots; Table 4 compares 0, 1, 5, and 10 shots, showing consistent improvement as the number of provided examples increased. The prompt strictly specified the output format, and the model outputs were parsed to extract trait name–score pairs, which were organized into a table with one row per essay. The total API cost was approximately USD 21, covering exploratory experiments, development dataset scoring, and test dataset scoring (around 21,346,941 tokens in total). The final version of the prompt template used for submission is included in the Appendix.

Shots	QWK Score
0-shot	0.579
1-shot	0.597
5-shot	0.603
10-shot	0.631

Table 4: Performance of few-shot prompting with varying numbers of examples during experiments.

- **Fine-Tuning BERT-Based Models:** Hyperparameter tuning explored adaptation scope {full, last-6, last-3 layers} and learning rates $\{1e{-}5, 2e{-}5, 3e{-}5\}$ under AdamW; training ran up to 100 epochs with early stopping on development macro-QWK.
- **Fine-Tuning Text-Generation LLM:** We fine-tuned `Sakalti/Saka-14B` with LoRA on all attention projections (r=32, α=64, dropout 0.08), using two rubric-aware supervision styles: *Simple CoT* (5–6 steps per trait) and *Advanced CoT* (7–8 steps per trait). Training ran on 5× NVIDIA TITAN RTX (24 GB) GPUs with learning rate $2e{-}5$, batch size 1, gradient accumulation 8, and fp16; we fixed the budget at **3 epochs** because training loss decreased monotonically, reaching **1.37** (Simple) and **0.31** (Advanced) by epoch 3, indicating continued fitting of the supervision. Inference used deterministic decoding (temperature 0.0, `max_new_tokens` 80), and outputs were parsed into seven trait-level integers and evaluated with QWK, MSE, and RMSE.

5 Results

This section presents system performance in both the development and testing phases. We report results using QWK, MSE, and RMSE across all traits. The analysis highlights the differences in agreement with human raters and calibration quality among all the models.

5.1 Development Phase

We first evaluated all four system families under the *cross-genre* setup (training on explanatory essays and testing on persuasive essays, and vice versa), averaging results across both directions. Table 5 reports the mean (QWK), mean squared error (MSE), and root mean squared error (RMSE) across the seven traits.

In Table 5, the "Fine-tuned BERT" row corresponds to **mDeBERTa-v3-base** trained with learning rate **2**e−**5**, **last-6 layers** unfrozen, with early stopping—the best single checkpoint among our BERT-based models—achieving the best development QWK *among our systems* (**0.575**), slightly below the shared-task baseline (**0.582**). GPT-based prompting (10 shots) was close (0.564), classical ML (AVG-pooling ridge with linguistic features) trailed (0.539), and the fine-tuned LLM lagged with lower QWK (0.480).

System	QWK	MSE	RMSE
Baseline	**0.582**	**0.504**	**0.699**
Fine-tuned BERT	0.575	0.596	0.758
GPT-based Few-Shot	0.564	0.549	0.727
Classical ML	0.539	0.624	0.777
Fine-tuned LLM	0.480	0.821	0.887

Table 5: Development set performance across system families.

5.2 Testing Phase

The official evaluation was conducted under a *cross-prompt* setting, where systems were tested on previously unseen prompts in challenging conditions. Table 6 reports macro-average results across all seven traits. GPT-based few-shot prompting achieved the strongest performance, improving from 0-shot (0.592 QWK) to 1-shot (0.610) and 10-shot (**0.612**), with GPT-1-shot also producing the lowest error rates (MSE **0.758**, RMSE **0.845**). Among non-GPT systems, **Classical ML** with AVG-pooling ridge and linguistic features reached 0.582 QWK, **Fine-tuned BERT** 0.554, and the **Fine-tuned LLM** 0.538; all exceeded the shared-task baseline (0.472).

Calibration differed by family. While GPT variants achieved both the highest QWK and the lowest error rates, **BERT** improved over the baseline on MSE (0.949 vs. 1.005) and RMSE (0.956 vs. 0.990) while maintaining moderate QWK. In contrast, **Classical ML** and the **Fine-tuned LLM** raised QWK but suffered from higher MSE (1.081 and 1.029, respectively). Taken together, these results suggest that GPT prompting is most effective for balancing *ordinal agreement* with human raters (QWK) and *absolute calibration* (MSE/RMSE), whereas other approaches achieve only partial and less consistent gains.

System	QWK	MSE	RMSE
Baseline	0.472	1.005	0.990
GPT-0-shot	0.592	0.797	0.867
GPT-1-shot	0.610	**0.758**	**0.845**
GPT-10-shot	**0.612**	0.760	0.848
Classical ML	0.582	1.081	1.038
Fine-tuned BERT	0.554	0.949	0.956
Fine-tuned LLM	0.538	1.029	0.995

Table 6: Official testing results.

5.3 Error Analysis

Across both development and testing phases, distinct error patterns emerged for each model family. **GPT few-shot** yields the highest exact-match rate, especially on *Relevance* and *Development*, with a mild tendency to under predict extremes. **BERT** systematically skews high, over predicting most on *Vocabulary*, *Style*, and *Grammar*. **Saka-14B (fine-tuned)** also overestimates, most visibly for *Vocabulary/Style*, and sporadically under predicts *Relevance*, indicating weaker calibration under unseen prompts. In contrast, the **Ridge** baseline consistently under predicts across traits, most notably for *Organization* and *Development*. Overall, GPT is best-calibrated, BERT/Saka tend to score high, and Ridge tends to score low, with these tendencies persisting from development (Figure 1) and testing (Figure 2).

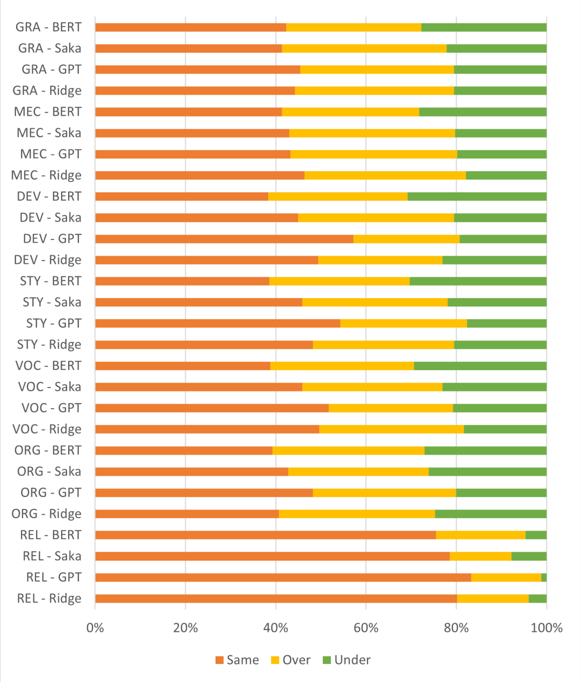


Figure 1: Development-phase calibration across traits and models. Bars show the proportion of predictions that were exact matches (*Same*), overestimates (*Over*), or underestimates (*Under*); stacks sum to 100%.

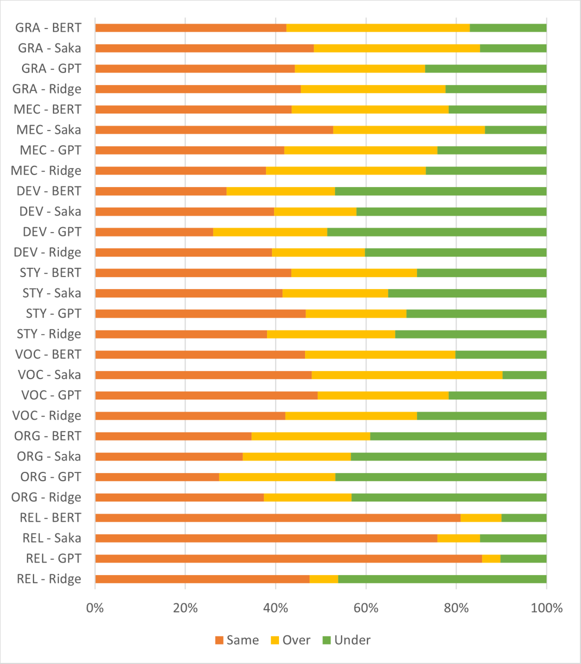


Figure 2: Testing-phase calibration across traits and models. Bars show the proportion of predictions that were exact matches (*Same*), overestimates (*Over*), or underestimates (*Under*); stacks sum to 100%.

6 Conclusion

In this study, we systematically compared multiple approaches to automated essay scoring, with particular emphasis on cross-genre generalization and alignment with trait-specific rubric criteria. By concatenating prompts, essays, and reasoning sequences with gold scores, our systems were explicitly encouraged to approximate human evaluation. Experimental results showed that fine-tuned BERT-based models achieved the highest QWK on the development set, slightly outperforming GPT-based few-shot prompting and classical ML approaches, while text-generation LLMs struggled under cross-genre conditions despite detailed CoT guidance.

The testing phase further demonstrated the robustness of GPT-based few-shot methods: providing in-context examples consistently improved performance, and translating rubrics and prompts into English enhanced trait calibration. Overall, this work shows that combining rubric-grounded reasoning with modern NLP architectures can yield reliable, trait-specific scoring of Arabic essays. These findings provide insights for practical deployment in educational contexts and point to future research directions focused on improving generalization, calibration, and interpretability in automated writing evaluation systems.

Acknowledgments

This work was supported by Cross-ministerial Strategic Innovation Promotion Program (SIP) on "Integrated Health Care System" Grant Number JPJ012425. We also extend our gratitude to the TAQEEM2025 organizers and reviewers for their valuable efforts in dataset preparation, rubric design, and evaluation. Their contributions were instrumental in enabling this research.

References

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Walid Massoud, Houda Bouamor, and Tamer Elsayed. 2025. TAQEEM 2025: Overview of the First Shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Eslam Zahran, Hager Elhuseyin, Tamer Elsayed, Walid Massoud, and Houda Bouamor. 2024. QAES: First Publicly-Available Trait-Specific Annotations for Automated Scoring of Arabic Essays. In *Proceedings of the Second Arabic Natural Language Processing Conference (ArabicNLP 2024)*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *arXiv preprint* arXiv:2111.09543.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv preprint* arXiv:1911.02116.

Go Inoue, Badr 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.

Sakalti. 2024. Saka-14B: An Arabic large language model. https://huggingface.co/Sakalti/Saka-14B.

Edward J. Hu, Yelong Shen, Phillip Wallis, Z. Allen-Zhu, Yuanzhi Li, Sébastien Bubeck, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint* arXiv:2106.09685.

Appendix

Prompt Specification for GPT Scoring

This section includes the final structured prompt template used in our GPT experiments, ensuring the reproducibility of our results.

```
"role": "system",
"content": "You are an expert Arabic language
    teacher responsible
for evaluating Arabic essays written by
    students based on specific
traits and rubrics."}

fixed_user_message = {
"role": "user",
"content": f"""Think step-by-step about the
    following criteria, then start scoring the
    provided essay:
- Essays are evaluated on the following traits:
    {traits}.
- Each trait is described in this rubric (the
    dictionary of each trait is
    score-explanation pairs): {rubric}.
- The essay was written in response to the
    following prompt: {essay_prompt}
- Essay type: {essay_type}
- A score of zero is given if the response is
    completely memorized, copied from the
    prompt,
  if the student did not attempt to complete
    the task, or wrote something unrelated to
    the required topic.

Scoring steps:
1. Check the trait and its rubric.
2. Read the essay.
3. Provide a score.
4. Repeat from step 1 for each trait.
5. After scoring all traits, format the output
    as follows:
<trait_name>: <score>
Do not provide any additional text or
    explanation.

Example essays with scores:
{examples}

"""}
```

GPT-Based Few-Shot Prompting

This section provides detailed results for GPT-based few-shot prompting. We report trait-level evaluation metrics for different shot settings, complementing the aggregate results in the main text.

Trait	QWK	MSE	RMSE
Relevance	0.545	0.170	0.411
Organization	0.712	0.800	0.894
Vocabulary	0.653	0.783	0.881
Style	0.620	0.981	0.986
Development	0.629	0.761	0.872
Mechanics	0.482	1.038	1.009
Grammar	0.506	1.048	1.014

Table 7: GPT-0-shot: Trait-level evaluation results.

Trait	QWK	MSE	RMSE
Relevance	0.585	0.158	0.395
Organization	0.711	0.802	0.894
Vocabulary	0.646	0.798	0.889
Style	0.666	0.841	0.914
Development	0.647	0.716	0.846
Mechanics	0.477	1.023	1.004
Grammar	0.544	0.969	0.972

Table 8: GPT-1-shot: Trait-level evaluation results.

Trait	QWK	MSE	RMSE
Relevance	0.553	0.168	0.406
Organization	0.709	0.821	0.905
Vocabulary	0.633	0.837	0.911
Style	0.654	0.863	0.926
Development	0.640	0.750	0.865
Mechanics	0.515	0.972	0.979
Grammar	0.580	0.908	0.944

Table 9: GPT-10-shot: Trait-level evaluation results.

Classical ML Features

This section lists the handcrafted linguistic features extracted from essays, which were combined with embeddings in the classical ML experiments.

Feature
Total words in the essay
Unique words in the essay
Punctuation marks count
Total sentences in the essay
Average word length (chars)
Average words per sentence
Total characters in the essay
Stopwords count
Total bigrams
Total trigrams
Unique bigrams
Unique trigrams
Unique/total bigrams ratio
Unique/total trigrams ratio

Table 10: Linguistic features extracted from essays.

912 at TAQEEM 2025: A Distribution-aware Approach to Arabic Essay Scoring

Trong-Tai Dam Vu, Dang Van Thin
University of Information Technology-VNUHCM,
Vietnam National University, Ho Chi Minh City, Vietnam
{taidvt,thindv}@uit.edu.vn

Abstract

We present our system for TAQEEM 2025 Task A on Arabic automatic essay scoring. Building on a pretrained Arabic encoder, our work focuses on two key design axes: (i) replacing the standard linear head with a lightweight multi-layer perceptron (MLP) and (ii) optimizing with distribution-aware objectives. We introduce a Weighted Mean-Squared Error loss, which assigns higher weights to less frequent scores to counteract the imbalanced, bell-shaped score distribution of the training data. On the official development folds, our system outperforms the baseline on Quadratic Weighted Kappa. Our findings underscore the importance of tailoring objective functions to specific data characteristics for achieving state-of-the-art results in AES.

1 Introduction

Automatic essay scoring (AES) aims to predict human-assigned holistic scores for free-form writing. The TAQEEM 2025 shared task focuses on Arabic AES (Task A), providing standardized data and an agreement-focused evaluation via QWK (Bashendy et al., 2025).

In line with the shared task guidelines, our goal is to conduct a transparent and reproducible study of what modifications yield reliable gains. Our work investigates two primary questions: 1) What is the optimal architecture for the prediction head? 2) Can a distribution-aware objective function, designed to address the specific characteristics of the score data, offer an advantage over standard regression losses?

Our system builds on the pretrained ArabicBERT v02 encoder (Antoun et al., 2020), which is based on the Transformer architecture (Vaswani et al., 2017; Devlin et al., 2019). We systematically explore the impact of MLP head depth and compare several objective functions. Our key contribution is the successful application of a Weighted MSE loss, which addresses the inherent imbalance in the dataset's score distribution. This simple, well-analyzed approach with a carefully chosen objective function can achieve state-of-the-art results.

2 Background

Task Description. TAQEEM 2025 is a shared task on evaluating Arabic student writing. We participated only in Task A (holistic AES). The official evaluation metric is QWK(Cohen, 1960).

The official dataset is composed of a training set of 426 Arabic essays and a test set of 840 essays, each covering two distinct writing prompts: explanatory and persuasive. The score distribution is bell-shaped and imbalanced toward mid-range scores, as shown in Figure 1. This observed imbalance is the primary motivation for our experiments with a weighted loss function, as standard MSE can be biased towards predicting the more frequent, mid-range scores.

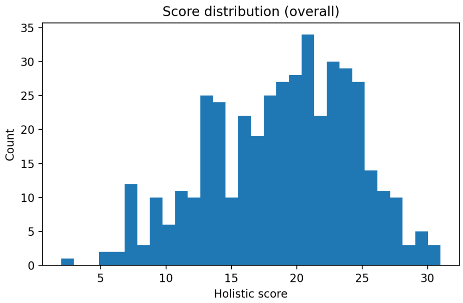


Figure 1: Overall distribution of holistic scores in the training set. The bell-shaped curve centered on mid-range scores (approx. 18-25) motivated our use of a weighted loss objective.

Related work. Automated Essay Scoring (AES) has evolved from early feature-based systems (Attali and Burstein, 2006) to deep learning. Currently, fine-tuning large pretrained Transformers like BERT (Devlin et al., 2019) is the state-of-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 983–988
November 8-9, 2025 ©2025 Association for Computational Linguistics

the-art approach, consistently achieving top results (Ludwig et al., 2021). Our work builds directly on this paradigm, leveraging a powerful pretrained Arabic model.

While much research focuses on English, Arabic AES is an active area (Ghazawi and Simpson, 2024), with models like AraBERT (Antoun et al., 2020) providing a strong foundation for the task. The paradigm of pre-training on large text corpora was popularized by both decoder-focused generative models like the GPT series (Radford et al., 2018, 2019; Brown et al., 2020) and encoder-focused models like BERT. Methodologically, our primary contribution—the use of a Weighted MSE loss—is inspired by established techniques for learning from imbalanced datasets (Cao et al., 2019; Ren et al., 2018), which we adapt to address the specific bell-shaped score distribution inherent in AES data.

3 System Overview

Our approach is centered on fine-tuning the AraBERTv02 model (Antoun et al., 2020). The overall architecture is depicted in Figure 2.

3.1 Backbone and Inputs

The core of our system is the AraBERT-v0.2 (Antoun et al., 2020) encoder, a pre-trained language model optimized for Arabic. To effectively present the task to the model, we explored two distinct input representations.

The first configuration, which we term Essay-only, provides the model with only the student's essay text. This approach tests the model's ability to infer scoring criteria directly from the text itself.

The second configuration, Essay and Prompt, uses a concatenation of the writing prompt and the essay text as input. This method provides the model with explicit context about the task's requirements. The choice between these two representations was determined empirically, as detailed in our ablation study.

3.2 Prediction Head

The standard approach for regression tasks with BERT-like models is to use a single linear layer (a regression head) on top of the `[CLS]` token representation. To explore if a more complex function could better map the learned features to a score, we experimented with replacing this linear head with a lightweight Multi-Layer Perceptron (MLP).

We systematically varied the depth of this MLP by changing the number of hidden layers, denoted by k. We tested configurations within the set $k \in \{0, 1, 2, 3\}$. The case where $k = 0$ is equivalent to the standard linear head, which serves as a direct baseline for this experiment. The optimal depth of the MLP was determined empirically, as we detail in our ablation studies.

3.3 Objectives

Our primary contribution in this work lies in the design and application of a distribution-aware objective function tailored to the specific characteristics of the AES dataset. We describe our proposed Weighted MSE (wMSE) loss below. To validate its effectiveness, we benchmarked it against the standard MSE loss and an agreement-aware MSE+QWK objective in our ablation studies.

Our proposed Weighted MSE (wMSE) loss is designed to counteract the imbalanced (Cao et al., 2019; Ren et al., 2018), bell-shaped score distribution of the training data. The core idea is to assign a weight, w_s, to each possible integer score s, where the weight is inversely proportional to the score's frequency in the training corpus $\mathcal{D}_{\text{train}}$. This forces the model to place greater importance on correctly predicting essays with rare scores.

First, for each unique integer score s in the range $[s_{\min}, s_{\max}]$, we calculate its frequency $N_s = |\{y_i \in \mathcal{D}_{\text{train}} \mid y_i = s\}|$. The weight w_s is then defined as the inverse of this frequency:

$$w_s = \frac{1}{N_s} \tag{1}$$

These weights are pre-calculated once over the entire training set. For a given batch of B samples, the Weighted MSE loss, $\mathcal{L}_{\text{wMSE}}$, is computed as the mean of the squared errors, where each error term is multiplied by the weight corresponding to its ground-truth label. For a prediction $\hat{y}_i$ and a true label y_i, the loss is:

$$\mathcal{L}_{\text{wMSE}} = \frac{1}{B} \sum_{i=1}^{B} w_{y_i} \cdot (\hat{y}_i - y_i)^2 \tag{2}$$

Since the ground-truth labels y_i are integers, the corresponding weight w_{y_i} can be retrieved directly.

To benchmark our proposed wMSE loss, we also evaluated two other objective functions. The standard **Mean Squared Error (MSE)** served as our main regression baseline. Additionally, we experimented with a combined **MSE+QWK** objective.

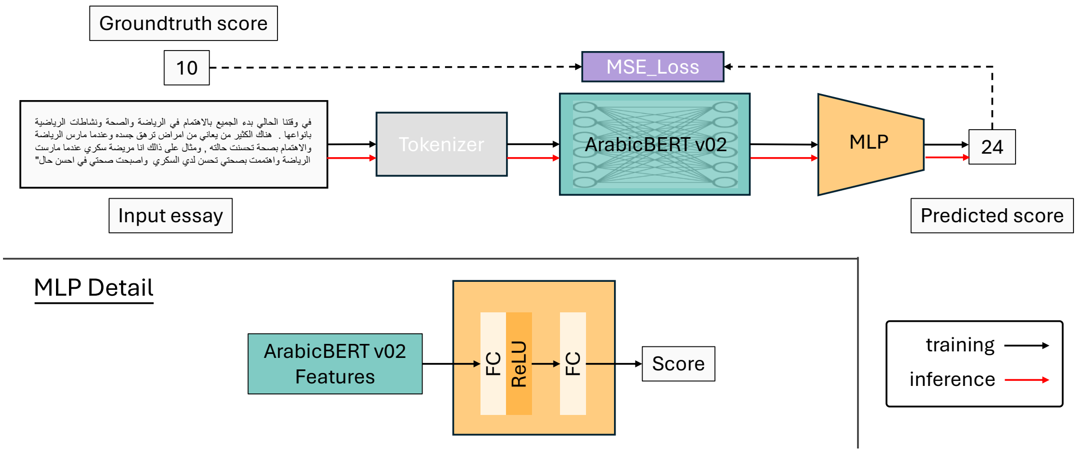

Groundtruth score
10
MSE_Loss
في وقتنا الحالي بدء الجميع بالاهتمام في الرياضة والصحة ونشاطات الرياضية
بانواعها . هناك الكثير من يعاني من امراض ترهق جسده وعندما مارس الرياضة
والاهتمام بصحة تحسنت حالته , ومثال على ذالك انا مريضة سكري عندما مارست
الرياضة واهتممت بصحتي تحسن لدي السكري واصبحت صحتي في احسن حال"
Tokenizer
ArabicBERT v02
MLP
24
Input essay
Predicted score
MLP Detail
ArabicBERT v02 Features
FC
ReLU
FC
Score
training
inference

Figure 2: Our system architecture, showing the ArabicBERT encoder followed by a one-layer MLP head for score prediction.

The MSE+QWK loss function incorporates a differentiable surrogate of Quadratic Weighted Kappa (QWK) directly into the training objective. This aims to align the model's optimization more closely with the final evaluation metric. Standard QWK is non-differentiable because it is calculated from a confusion matrix, which requires rounding the model's continuous regression outputs (e.g., 13.7) into discrete integer predictions (e.g., 14). This rounding step prevents gradients from flowing during backpropagation.

To create a differentiable surrogate, we implemented a "soft" version of the QWK calculation. The process begins with soft assignment, where instead of rounding, each continuous prediction is represented as a soft probability distribution over all possible integer scores. This is achieved by calculating the distance from the prediction to each integer class center (e.g., the distances from 13.7 to) and converting these distances into a probability vector using a softmax function. A prediction of 13.7 will thus have high probabilities assigned to the nearby classes 13 and 14. This process is also applied to the ground-truth labels, naturally handling non-integer scores. These resulting probability vectors are then used to construct a "soft" confusion matrix for the batch by summing the outer product of each prediction-label vector pair. With this fully differentiable confusion matrix, the observed and expected agreement can be calculated using standard matrix operations, allowing gradients to flow back through the entire QWK formula to the model's outputs.

The combined loss is then formally defined as:

$$\mathcal{L}_{\text{MSE+QWK}} = \mathcal{L}_{\text{MSE}} + (1 - \text{QWK}) \qquad (3)$$

where QWK is the fully differentiable surrogate of the QWK metric, calculated as described above.

4 Experimental Setup

To ensure reproducibility and isolate the impact of our design choices, we conducted a systematic ablation study. All models were fine-tuned using the AdamW optimizer with a learning rate of 5e-5, a batch size of 16, for up to 100 epochs. The best checkpoint for each run was selected based on the highest average QWK on the development folds. Our study evaluated three primary design axes: 1) the objective function (our proposed Weighted MSE vs. standard MSE and MSE+QWK), 2) the MLP head architecture (varying the number of hidden layers $k \in \{0, 1, 2, 3\}$), and 3) the input type (essay-only vs. prompt+essay). The results presented in Table 1 compare the best-performing configuration found for each objective to ensure a fair and comprehensive analysis.

5 Results and Analysis

Our main experimental results are summarized in Table 1, which presents a comprehensive ablation study. The final official scores on the private test set are shown in Table 2.

5.1 Overall Performance

Our best single model achieved an average QWK of **0.766** on the development set (0.784 on Fold 1 and 0.747 on Fold 2). As shown in Table 2, our

Configuration	Loss	Input	MLP Depth (k)	Dev QWK (Fold 1)	Dev QWK (Fold 2)	Avg QWK
Baseline (Linear Head)	MSE	Essay	0	0.705	0.727	0.716
Our Best Model	**Weighted MSE**	**Essay**	**1**	**0.784**	0.747	**0.766**
Ablation on Objective Function						
- use MSE Loss	MSE	Essay	3	0.768	**0.753**	0.761
- use MSE+QWK Loss	MSE+QWK	Essay	2	0.741	0.752	0.747
Ablation on Architecture						
- use 2 hidden layers	Weighted MSE	Essay	2	0.768	0.752	0.760
- use 3 hidden layers	Weighted MSE	Essay	3	0.781	0.740	0.761
Ablation on Input Type						
- use Prompt+Essay	Weighted MSE	Prompt+Essay	1	0.764	**0.753**	0.759

Table 1: Main results and a comprehensive ablation study on the development set. Performance is measured by Quadratic Weighted Kappa (QWK), averaged over two folds. The table compares our best model (in bold) against the official baseline. It also presents three sets of ablation studies, each starting from our best model's configuration and varying a single component: the objective function, architecture, or input type.

Configuration	**QWK (Fold 9)**	**QWK (Fold 10)**	**Official QWK**	**Official RMSE**
Baseline	0.608	0.670	0.639	5.372
Our Best Model	**0.662**	**0.683**	**0.673**	**5.333**

Table 2: Final performance on the private test set, comparing our best model to the official baseline. We report the official QWK and RMSE, along with the QWK scores from the last two cross-validation folds.

best model significantly outperforms the baseline on the private test set, confirming the effectiveness of our approach on unseen data.

5.2 Analysis of Findings

Our ablation studies, detailed in Table 1, provide several key insights into the factors driving performance.

Impact of Objective Function. The choice of objective function is the most critical factor for success. Our **Weighted MSE** model (Avg QWK 0.766) significantly outperforms the best configurations using standard MSE (0.761) and MSE+QWK (0.747). This confirms our hypothesis that explicitly addressing the dataset's imbalanced score distribution is crucial for achieving top performance. By forcing the model to pay more attention to less frequent scores, the wMSE objective mitigates the model's natural bias towards the populated mean of the distribution.

Interplay between Architecture and Objective. The architectural ablation study reveals a clear relationship between our proposed wMSE objective and the model's architectural complexity. As shown in Table 1, the performance of the wMSE-trained model peaks with a 1-layer MLP ($k = 1$). Performance degrades when the architecture is too simple ($k = 0$, a standard linear head) and also when it becomes overly complex ($k = 2, 3$).

This suggests that the wMSE loss, by increasing the importance of rare scores, creates a more challenging optimization landscape than standard MSE. A simple linear head ($k = 0$) appears to lack sufficient capacity to fully model the nuances of this distribution-aware objective. Conversely, deeper MLPs ($k = 2, 3$) seem prone to overfitting on this specialized task. Therefore, our results indicate that the benefits of a distribution-aware objective are best realized when paired with an architecture of appropriate, non-trivial complexity.

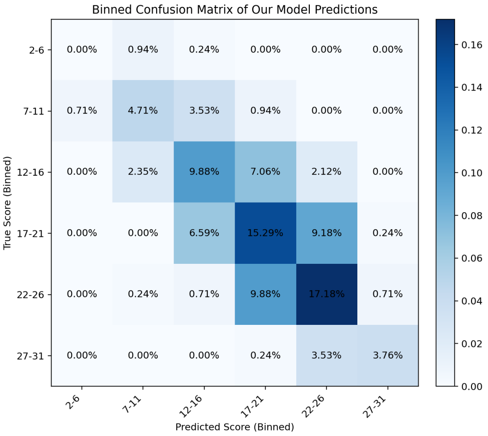
Impact of Input Type. The ablation on input type confirms that an essay-only approach is optimal for our best model. Including the prompt text resulted in a performance drop (from 0.766 to 0.759 Avg QWK). While the prompt provides essential context for evaluating aspects like relevance, our empirical results suggest that its explicit inclusion via concatenation is suboptimal in this setup. We hypothesize two potential reasons for this counterintuitive finding. First, since the dataset contains only two distinct prompts, the model may be able to implicitly infer the necessary context from the essay's topic, vocabulary, and structure alone, making the explicit prompt text redundant. Second, concatenating the prompt might unfavorably shift the model's attentional focus. The model may allocate too much of its limited attention capacity to the initial prompt tokens, thereby diluting its focus on the nuanced linguistic features distributed throughout the essay itself.

5.3 Error Analysis

To better understand our model's limitations, we analyzed its prediction errors on the development set. Figure 3 presents a binned confusion matrix of our best model's predictions, which visually confirms our two primary failure modes:

1. **Near-Boundary Confusion:** The strong concentration of predictions along the main diagonal and its adjacent cells is the most prominent pattern. This shows that the model's primary error is confusing similar, adjacent score ranges (e.g., predicting a score in the 17-21 bin for a true score in the 22-26 bin). This is a classic challenge in regression-based AES.

2. **Off-Prompt Responses:** The dataset contains some essays that do not fully address the prompt. Our model, trained on holistic writing quality, sometimes assigns a moderate score to a well-written but off-topic essay, whereas a human grader might penalize it more heavily for being non-responsive to the task.



	2-6	7-11	12-16	17-21	22-26	27-31
2-6	0.00%	0.94%	0.24%	0.00%	0.00%	0.00%
7-11	0.71%	4.71%	3.53%	0.94%	0.00%	0.00%
12-16	0.00%	2.35%	9.88%	7.06%	2.12%	0.00%
17-21	0.00%	0.00%	6.59%	15.29%	9.18%	0.24%
22-26	0.00%	0.24%	0.71%	9.88%	17.18%	0.71%
27-31	0.00%	0.00%	0.00%	0.24%	3.53%	3.76%

Figure 3: Binned confusion matrix of predictions on the development set. The concentration of values around the main diagonal highlights near-boundary confusion as the primary error type.

6 Conclusion

We presented our system for the TAQEEM 2025 Task A on Arabic AES. Our success was primarily driven by a custom Weighted MSE (wMSE) objective, designed to counteract the imbalanced, bell-shaped score distribution of the training data. Our analysis revealed a crucial finding: this distribution-aware objective not only significantly boosted performance but also achieved its best results with a simpler 1-layer MLP architecture compared to the deeper models required by standard MSE. Our work underscores the value of tailoring objective functions to data characteristics and demonstrates that a simple, well-analyzed approach can achieve state-of-the-art results in AES.

7 Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

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 (OSACT)*.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Walid Massoud, Houda Bouamor, and Tamer Elsayed. 2025. TAQEEM 2025: Overview of the First Shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in neural information processing systems (NeurIPS)*, volume 33, pages 1877–1901.

Kaidi Cao, Chen Wei, Adrien Gaidon, Niki Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

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)*.

Rayed Ghazawi and Edwin Simpson. 2024. Automated essay scoring in arabic: a dataset and analysis of a bert-based system. *arXiv preprint arXiv:2407.11212*.

Sabrina Ludwig, Christian Mayer, Christopher Hansen, Kerstin Eilers, and Steffen Brandt. 2021. Automated essay scoring using transformer models. *Psych*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning (ICML)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*.

Taibah at TAQEEM 2025: Leveraging GPT-4o for Arabic Essay Scoring

NADA ALMARWANI[1], **ALAA ALHARBI**[2], **SAMAH ALOUFI**[1]
[1]Department of AI and Data Science, CCSE, Taibah University
[2]Department of Information Systems, CCSE, Taibah University

(e-mail: nmarwani, alaharbi, slhebi@taibahu.edu.sa)

Abstract

This paper presents our system submitted to TAQEEM 2025, which designed to address two tasks: (A) holistic scoring and (B) trait-specific scoring. We propose a GPT-4o-based methodology that employs few-shot prompting to serve as a grader for both tasks. Specifically, for task A, we utilize prompt-based scoring criteria with exemplars to assess overall essay quality. For task B, we design trait-specific prompting schemes to capture fine-grained grading aspects. Our system attains substantial agreement on Task A (QWK = 0.75) and a mean QWK of 0.65 across traits for task B, outperforming the shared task baseline on both tasks.

1 Introduction

Evaluating student essays plays a critical role in assessing language proficiency and writing development, particularly in educational settings where writing is a core skill. However, traditional essay scoring is labor-intensive, costly, and liable to inter- and intra-rater inconsistencies caused by human subjectivity, bias, and rater characteristics such as severity or leniency (Uto and Okano, 2020). To address these challenges, Automated Essay Scoring (AES) systems have emerged as scalable and efficient alternatives. When effectively implemented, AES systems offer timely, objective, and consistent scoring, mitigating rater bias and supporting large-scale assessment contexts such as standardized examinations.

Recent advancements in natural language processing (NLP), particularly the emergence of generative large language models (LLMs) such as OpenAI's GPT-4 and Google's PaLM, have significantly enhanced the capabilities of AES systems. A notable advantage of LLMs is their ability to perform zero-shot and few-shot scoring with minimal supervision. Mizumoto and Eguchi (2023) demonstrated that generative models like ChatGPT can reliably assess essays using standardized rubrics, confirming their feasibility and effectiveness for AES tasks. In terms of validity and reliability, Pack et al. (2024) and Li and Liu (2024) showed that GPT-4 achieved substantial agreement with human raters on AES tasks. Moreover, LLMs can be prompted to evaluate essays either via traditional linguistic features or rubric-based criteria aligned with human judgment (Pack et al., 2024). Recent work highlights that prompting strategies play a critical role in aligning LLM-generated scores with human evaluations (Li and Liu, 2024; Liew and Tan, 2024).

The majority of studies that have exploited LLMs for essay scoring have concentrated on English-language essays (Pack et al., 2024; Liew and Tan, 2024; Yavuz et al., 2025; Katuka et al., 2024; Yang, 2024; Flodén, 2025), with limited studies exploring other languages such as Chinese (Feng et al., 2024), Japanese (Li and Liu, 2024), and Arabic (Ghazawi and Simpson, 2025). The scarcity of annotated essay datasets in Arabic, which hinders the development of effective AES systems for this language, reflects a broader challenge. To address this gap, the TAQEEM shared task[1] (Bashendy et al., 2025) invites researchers to develop automated scoring models for Arabic essays, evaluating both holistic and trait-specific performance in a cross-prompt setting. Inspired by the promising results of prior work on generative LLM-based essay scoring, we employ OpenAI's GPT-4o model to simulate expert grading of Arabic essays across both tasks. Our approach leverages carefully crafted rubric-guided prompts and few-shot exemplars to achieve consistent and interpretable scoring across diverse Arabic texts. We also conduct a concise error analysis quantifying over- and under-scoring.

[1] https://sites.google.com/view/taqeem-2025/home?authuser=0

Proceedings of The Third Arabic Natural Language Processing Conference, pages 989–997
November 8-9, 2025 ©2025 Association for Computational Linguistics

2 Task Description

The TAQEEM benchmark aims to advance automated Arabic essay scoring under cross-prompt evaluation via two tasks.

Task A: (Holistic Scoring) requires a single score reflecting the overall essay quality. **Task B: (Trait-specific Scoring)** requires the model to produce a separate score for seven rubric traits: Relevance, Organization, Vocabulary, Style, Development, Mechanics, and Grammar. The dataset provided with the TAQEEM 2025 Shared task comprises 1,265 Arabic essays, divided into 425 essays for training and 840 for testing. Each essay was written in response to one of several prompts and annotated by human for both tasks. This setup assesses systems' ability to generalize across prompts while maintaining alignment with human judgments.

3 Methodology

The essay grading system developed in this study leverages OpenAI's GPT-4o model (Hurst et al., 2024) to simulate expert scoring of Arabic essays. A small set of human-scored examples—specifically, 20 representative training samples—is embedded directly in the prompt as exemplars to guide the model through the grading process, ensuring coverage of the full range of grades. These 20 examples are randomly selected from the training dataset across a range of score levels to ensure diversity and enhance the model's ability to generalize across varying levels of essay quality, while remaining within token constraints. Importantly, all 20 exemplars were sourced from a single training prompt. These examples are then used to evaluate the model on different prompts. This checks if the model can perform well beyond the specific prompt on which it was trained, showing its adaptability across various inputs.

For Task A, the prompt includes a rubric for evaluating essays written in Arabic. This rubric assesses six core dimensions: content clarity, linguistic correctness, structural organization, strength of arguments, stylistic quality, and adherence to word count requirements. The dimensions were derived directly from the task description to ensure relevance, rather than adopting the CAST rubric, which may not have aligned with the task's unique requirements. These evaluation criteria are expressed in natural language instructions, enabling the model to internalize the scoring logic without relying on a structured input format. The original Arabic prompt, its English translation, and the rubric structure are provided in Figure 1 in the Appendix.

For Task B, we designed a structured prompt that also guides the model in evaluating Arabic student essays, simulating the behavior of an expert Arabic language teacher. This prompt instructs the model to score essays according to a detailed, criterion-referenced rubric covering seven dimensions: Relevance (max 2 points), Organization (max 5 points), Vocabulary (max 5 points), Style (max 5 points), Development (max 5 points), Mechanics (max 5 points), and Grammar (max 5 points)[2]. Each dimension is defined in natural language to ensure interpretability and consistent application of the scoring criteria. The original Arabic version of the prompt, as well as its English translation and associated rubric, are provided in Figure 2 in the Appendix.

The grading process is executed using OpenAI's API. Each essay, along with its corresponding prompt, is submitted to the GPT-4o model with a low temperature setting (0.1) to produce consistent and deterministic output.

4 Results

This section presents the performance comparison between the baseline system, which fine-tunes AraBERTv02 (Antoun et al., 2020) for automated essay scoring[3], and the proposed Taibah system for Task A and Task B, as detailed in Table 1 and Table 2, respectively. The evaluation was conducted using three key metrics: Quadratic Weighted Kappa (QWK), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

4.1 Task A: Holistic Scoring Results

As shown in Table 1, the Taibah system consistently outperforms the baseline in terms of QWK. For Test Prompt 9, our system achieved a QWK of 0.717, compared to the baseline's 0.608. This performance advantage remains evident in Test Prompt 10, where the QWK reached 0.784 versus the baseline's 0.670. The average QWK across both prompts was 0.751 for our system, demonstrating a notable improvement over the baseline's average of 0.639 and indicating stronger alignment with human judgments. This +0.112 increase in QWK reflects a substantial gain in rater agreement,

[2] https://sites.google.com/view/taqeem-2025

[3] https://gitlab.com/bigirqu/taqeem2025

System	Prompt 9			Prompt 10			Average		
	QWK	MSE	RMSE	QWK	MSE	RMSE	QWK	MSE	RMSE
Baseline	0.608	33.148	5.757	0.670	24.862	4.986	0.639	29.005	5.372
Taibah	0.717	31.281	5.593	0.784	19.595	4.427	**0.751**	**25.438**	**5.010**

Table 1: Performance comparison between Baseline and Taibah system for Task A. **Bold values** indicate superior performance.

especially considering that QWK values above 0.75 are often interpreted as indicating substantial to near-perfect agreement (Landis and Koch, 1977). We attribute this improvement, in part, to the rubric-aligned prompt design and the inclusion of diverse exemplars, which helped guide the model's scoring decisions.

In terms of error metrics, where lower values indicate better performance, the Taibah system also demonstrated superior performance. For Test Prompt 9, it achieved an MSE of 31.281 and RMSE of 5.593, outperforming the baseline's MSE of 33.148 and RMSE of 5.757. The advantage was even more pronounced for Test Prompt 10, with our system achieving an MSE of 19.595 and RMSE of 4.427 compared to the baseline's 24.862 (MSE) and 4.986 (RMSE). On average, the Taibah system maintained lower error rates (MSE: 25.438; RMSE: 5.010) than the baseline (MSE: 29.005; RMSE: 5.372), further validating its enhanced performance in Task A. These consistent reductions in MSE and RMSE across both prompts suggest that the few-shot GPT-4o-based approach generalizes well across different essay topics, despite the cross-prompt evaluation setting.

4.2 Task B: Trait-specific Scoring Results

As shown in Table 2, our system consistently outperformed the baseline across all traits and both test prompts in terms of QWK, demonstrating stronger alignment with human judgments in trait-level scoring. For Prompt 9, the most notable improvements were observed in Relevance (Taibah: 0.586 vs. Baseline: 0.127) and Development (Taibah: 0.727 vs. Baseline: 0.410). Similar trends were seen for Prompt 10, with substantial gains in Relevance (Taibah: 0.538 vs. Baseline: 0.182) and Mechanics (Taibah: 0.686 vs. Baseline: 0.468). On average, our system achieved higher QWK scores across all traits, with the largest improvements in Development (Taibah: 0.703 vs. Baseline: 0.458) and Relevance (Taibah: 0.562 vs. Baseline: 0.155). These gains are particularly important for scoring dimensions that are often challenging for automated systems, such as content relevance and argument development, suggesting that the prompt structure effectively guided the model's understanding of nuanced writing features.

Our system also demonstrated superior performance in error metrics, with lower MSE and RMSE values indicating better predictive accuracy. For Prompt 9, the system achieved notable reductions in both metrics across all traits. For example, in Relevance, MSE dropped from 0.514 (Baseline) to 0.221 (Taibah), and RMSE from 0.717 to 0.471. Similar improvements were observed in Development, where MSE decreased from 1.174 to 0.717 and RMSE from 1.083 to 0.847.

For Prompt 10, the system maintained its performance advantage. In Relevance, MSE decreased from 0.340 to 0.231 and RMSE from 0.584 to 0.481. Vocabulary also saw notable reductions, with MSE dropping from 0.964 to 0.669 and RMSE from 0.982 to 0.818. These results reflect the model's ability to generalize across writing prompts, a key challenge in cross-prompt AES settings.

Overall, our system achieved consistently lower average MSE and RMSE values across all traits. The most significant reductions were found in Relevance (MSE: 0.427 Baseline vs. 0.226 Taibah; RMSE: 0.651 baseline vs. 0.476 Taibah) and Vocabulary (MSE: 1.031 Baseline vs. 0.795 Taibah; RMSE: 1.015 Baseline vs. 0.889 Taibah). These findings reinforce the system's enhanced accuracy and reliability in trait-specific scoring for Task B, particularly for dimensions that require deeper semantic understanding.

5 Error Analysis and Discussion

Figure 5 in the Appendix presents confusion-matrix heatmaps for the test set that summarize prediction errors for Task A. Across both prompts, predicted scores concentrate around a few values such as 14, 18, 24, and 28 which leads to over-scoring of low-quality essays and under-scoring of high-

System	Trait	Prompt 9			Prompt 10			Average		
		QWK	MSE	RMSE	QWK	MSE	RMSE	QWK	MSE	RMSE
Relevance	Baseline	0.127	0.514	0.717	0.182	0.340	0.584	0.155	0.427	0.651
	Taibah	0.586	0.221	0.471	0.538	0.231	0.481	**0.562**	**0.226**	**0.476**
Organization	Baseline	0.563	1.117	1.057	0.619	0.954	0.962	0.591	1.036	1.010
	Taibah	0.680	0.945	0.972	0.656	0.948	0.973	**0.668**	**0.947**	**0.973**
Vocabulary	Baseline	0.546	1.098	1.048	0.602	0.964	0.982	0.574	1.031	1.015
	Taibah	0.609	0.921	0.960	0.675	0.669	0.818	**0.642**	**0.795**	**0.889**
Style	Baseline	0.560	1.164	1.079	0.584	0.981	0.990	0.572	1.073	1.035
	Taibah	0.662	0.960	0.980	0.693	0.748	0.865	**0.678**	**0.854**	**0.923**
Development	Baseline	0.410	1.174	1.083	0.506	0.883	0.940	0.458	1.029	1.012
	Taibah	0.727	0.717	0.847	0.679	0.795	0.892	**0.703**	**0.756**	**0.870**
Mechanics	Baseline	0.421	1.345	1.160	0.468	1.212	1.101	0.445	1.279	1.131
	Taibah	0.602	1.033	1.017	0.686	0.719	0.848	**0.644**	**0.876**	**0.933**
Grammar	Baseline	0.494	1.243	1.115	0.532	1.079	1.039	0.513	1.161	1.077
	Taibah	0.629	1.036	1.018	0.699	0.721	0.849	**0.664**	**0.879**	**0.934**

Table 2: Performance comparison between Baseline and Taibah system for Task B. **Bold values** indicate superior performance.

quality essays. Predictions at the extreme values 0–2 and 30–32 are rare even when the true scores lie in those ranges. In few cases, essays with a true score of 0 receive mid-range predictions which indicate leniency toward severely deficient responses. Two factors may contribute to this: the training data may contain few or no essays labeled 0, and the scoring instruction used in the prompting specified a 1–32 range rather than 0–32 which can drive predictions away from 0. Most errors lie within ±3 points (±1 to ±3 points). The Pearson correlation between human and model scores is high ($r = 0.87$), indicating overall agreement despite systematic bias. Here, we define *bias* as the signed difference between model and human scores: $\Delta =$ model $-$ human; $\Delta < 0$ indicates underestimation and $\Delta > 0$ indicates over-scoring. Essays with human scores ≥ 26 are most often underestimated. On average, the model underestimates relative to human ratings by about 0.73 points, a statistically significant difference ($t = -2.53$, $p = 0.012$). Figure 3 (Appendix) illustrates this pattern: the model is more lenient at the lower end of the scale and increasingly conservative at the upper end.

Furthermore, analysis of essay length indicates that very short essays with 0–50 words yield poor performance. Performance improves with length and peaks around 150–200 words. Beyond ≈ 300 words, the MAE increases even as QWK increases, suggesting that the system preserves ranking but tends to over-score longer texts. Very long essays that exceed 500 words show low agreement and large errors.

For Task B, the model tends to assign lower scores compared to human raters for Development, Style and Organization, with the largest mean biases in the Development (-0.254) and Style (-0.225), both highly significant ($p < 0.0001$). Vocabulary is the only trait with a small positive bias ($+0.088$, $p = 0.004$), while Mechanics shows no significant difference ($p = 0.14$). For relevance, which is scored on a scale of 0–2, the observed QWK score of 0.56 is reasonable given the narrow range. Figure 4 in the Appendix shows the mean bias ($Model - Human$) for each trait at each human score level. The model tends to over-scores the lowest-performing essays and underestimate high-scoring ones, leading to more negative bias toward the upper end of the human score scale.

6 Conclusion

This study presented our proposed system for automated Arabic essay scoring which submitted to TAQEEM 2025 shared task. The system leverages GPT-4o with a few-shot prompting methodology to evaluate the quality of Arabic essays. Our system achieved strong overall performance in both holistic scoring and trait-specific scoring tasks. For future work, we aim to enhance the system scalability and generalizability by expanding the dataset to encompass a broader range of topics and writing traits.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Walid Massoud, Houda Bouamor, and Tamer Elsayed. 2025. TAQEEM 2025: Overview of the First Shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

Haiyue Feng, Sixuan Du, Gaoxia Zhu, Yan Zou, Poh Boon Phua, Yuhong Feng, Haoming Zhong, Zhiqi Shen, and Siyuan Liu. 2024. Leveraging large language models for automated chinese essay scoring. In *International Conference on Artificial Intelligence in Education*, pages 454–467. Springer.

Jonas Flodén. 2025. Grading exams using large language models: A comparison between human and ai grading of exams in higher education using chatgpt. *British educational research journal*, 51(1):201–224.

Rayed Ghazawi and Edwin Simpson. 2025. How well can llms grade essays in arabic? *arXiv preprint arXiv:2501.16516*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Gloria Ashiya Katuka, Alexander Gain, and Yen-Yun Yu. 2024. Investigating automatic scoring and feedback using large language models. *arXiv preprint arXiv:2405.00602*.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Wenchao Li and Haitao Liu. 2024. Applying large language models for automated essay scoring for non-native japanese. *Humanities and Social Sciences Communications*, 11(1):1–15.

Pei Yee Liew and Ian KT Tan. 2024. On automated essay grading using large language models. In *Proceedings of the 2024 8th International Conference on Computer Science and Artificial Intelligence*, pages 204–211.

Atsushi Mizumoto and Masaki Eguchi. 2023. Exploring the potential of using an ai language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050.

Austin Pack, Alex Barrett, and Juan Escalante. 2024. Large language models and automated essay scoring of english language learner writing: Insights into validity and reliability. *Computers and Education: Artificial Intelligence*, 6:100234.

Masaki Uto and Masashi Okano. 2020. Robust neural automated essay scoring using item response theory. In *International conference on artificial intelligence in education*, pages 549–561. Springer.

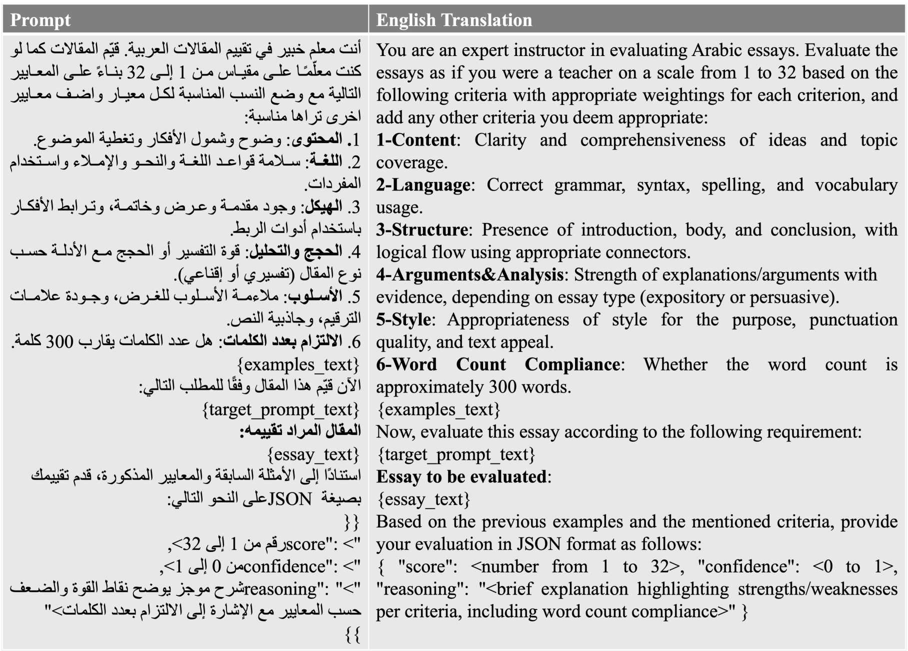
Yang Yang. 2024. The reliability of using chatgpt in rating efl writings. *Shanlax International Journal of Education*, 12(4):49–59.

Fatih Yavuz, Özgür Çelik, and Gamze Yavaş Çelik. 2025. Utilizing large language models for efl essay grading: An examination of reliability and validity in rubric-based assessments. *British Journal of Educational Technology*, 56(1):150–166.

A Appendix

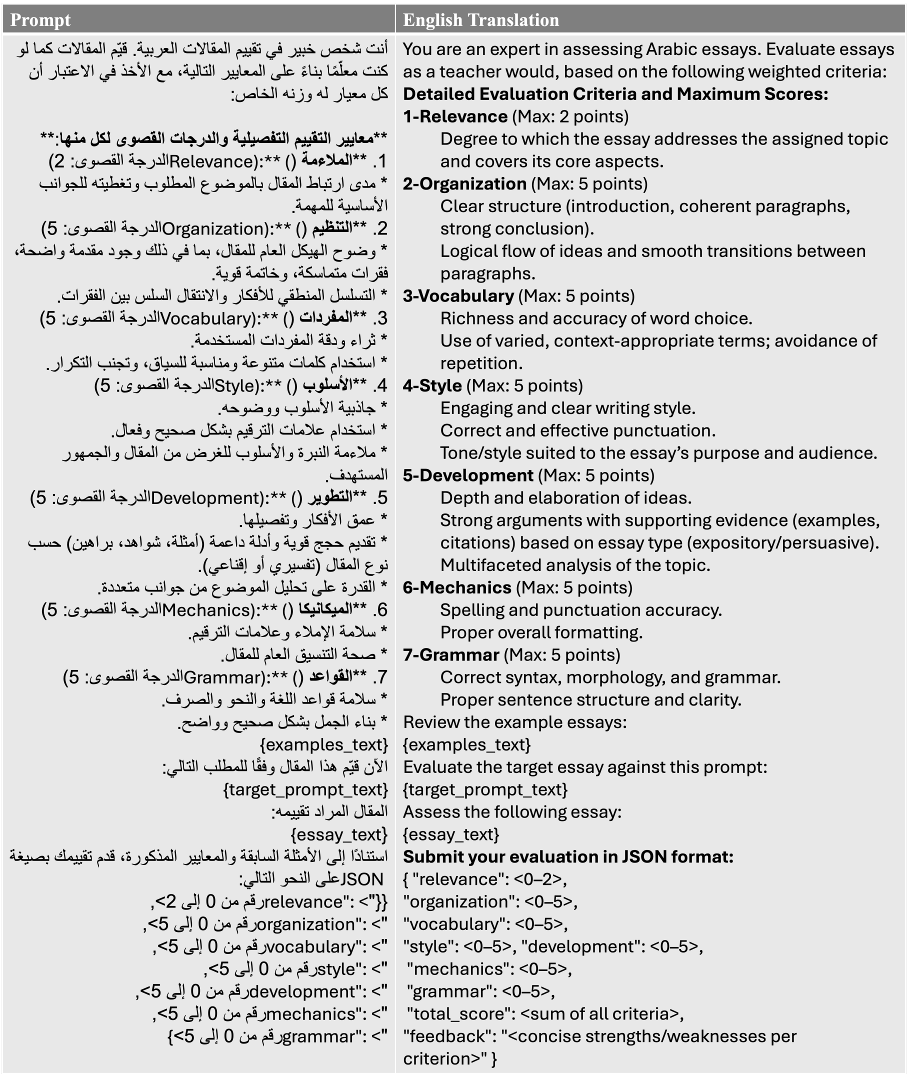
A.1 Structured Prompt Templates for Automated Essay Scoring

Figure 1 shows the structured prompt template used in Task A's automated essay scoring system. Figure 2 displays the structured prompt template applied in Task B's automated essay scoring system.



Prompt	English Translation
أنت معلم خبير في تقييم المقالات العربية. قيّم المقالات كما لو كنت معلّمًا على مقياس من 1 إلى 32 بناءً على المعايير التالية مع وضع النسب المناسبة لكل معيار واضف معايير اخرى تراها مناسبة: 1. **المحتوى**: وضوح وشمول الأفكار وتغطية الموضوع. 2. **اللغة**: سلامة قواعد اللغة والنحو والإملاء واستخدام المفردات. 3. **الهيكل**: وجود مقدمة وعرض وخاتمة، وترابط الأفكار باستخدام أدوات الربط. 4. **الحجج والتحليل**: قوة التفسير أو الحجج مع الأدلة حسب نوع المقال (تفسيري أو إقناعي). 5. **الأسلوب**: ملاءمة الأسلوب للغرض، وجودة علامات الترقيم، وجاذبية النص. 6. **الالتزام بعدد الكلمات**: هل عدد الكلمات يقارب 300 كلمة. {examples_text} الآن قيّم هذا المقال وفقًا للمطلب التالي: {target_prompt_text} **المقال المراد تقييمه:** {essay_text} استنادًا إلى الأمثلة السابقة والمعايير المذكورة، قدم تقييمك بصيغة JSONعلى النحو التالي: {{ "score": <"رقم من 1 إلى 32>, "confidence": <"من 0 إلى 1>, "reasoning": "<"شرح موجز يوضح نقاط القوة والضعف حسب المعايير مع الإشارة إلى الالتزام بعدد الكلمات>" }}	You are an expert instructor in evaluating Arabic essays. Evaluate the essays as if you were a teacher on a scale from 1 to 32 based on the following criteria with appropriate weightings for each criterion, and add any other criteria you deem appropriate: **1-Content**: Clarity and comprehensiveness of ideas and topic coverage. **2-Language**: Correct grammar, syntax, spelling, and vocabulary usage. **3-Structure**: Presence of introduction, body, and conclusion, with logical flow using appropriate connectors. **4-Arguments&Analysis**: Strength of explanations/arguments with evidence, depending on essay type (expository or persuasive). **5-Style**: Appropriateness of style for the purpose, punctuation quality, and text appeal. **6-Word Count Compliance**: Whether the word count is approximately 300 words. {examples_text} Now, evaluate this essay according to the following requirement: {target_prompt_text} **Essay to be evaluated**: {essay_text} Based on the previous examples and the mentioned criteria, provide your evaluation in JSON format as follows: { "score": <number from 1 to 32>, "confidence": <0 to 1>, "reasoning": "<brief explanation highlighting strengths/weaknesses per criteria, including word count compliance>" }

Figure 1: Structured Prompt Template Applied in Task A's Automated Essay Scoring System.



Prompt	English Translation
أنت شخص خبير في تقييم المقالات العربية. قيّم المقالات كما لو كنت معلّمًا بناءً على المعايير التالية، مع الأخذ في الاعتبار أن كل معيار له وزنه الخاص: **معايير التقييم التفصيلية والدرجات القصوى لكل منها:** 1. **الملاءمة () **:(Relevanceالدرجة القصوى: 2) * مدى ارتباط المقال بالموضوع المطلوب وتغطيته للجوانب الأساسية للمهمة. 2. **التنظيم () **:(Organizationالدرجة القصوى: 5) * وضوح الهيكل العام للمقال، بما في ذلك وجود مقدمة واضحة، فقرات متماسكة، وخاتمة قوية. * التسلسل المنطقي للأفكار والانتقال السلس بين الفقرات. 3. **المفردات () **:(Vocabularyالدرجة القصوى: 5) * ثراء ودقة المفردات المستخدمة. * استخدام كلمات متنوعة ومناسبة للسياق، وتجنب التكرار. 4. **الأسلوب () **:(Styleالدرجة القصوى: 5) * جاذبية الأسلوب ووضوحه. * استخدام علامات الترقيم بشكل صحيح وفعال. * ملاءمة النبرة والأسلوب للغرض من المقال والجمهور المستهدف. 5. **التطوير () **:(Developmentالدرجة القصوى: 5) * عمق الأفكار وتفصيلها. * تقديم حجج قوية وأدلة داعمة (أمثلة، شواهد، براهين) حسب نوع المقال (تفسيري أو إقناعي). * القدرة على تحليل الموضوع من جوانب متعددة. 6. **الميكانيكا () **:(Mechanicsالدرجة القصوى: 5) * سلامة الإملاء وعلامات الترقيم. * صحة التنسيق العام للمقال. 7. **القواعد () **:(Grammarالدرجة القصوى: 5) * سلامة قواعد اللغة والنحو والصرف. * بناء الجمل بشكل صحيح وواضح. {examples_text} الآن قيّم هذا المقال وفقًا للمطلب التالي: {target_prompt_text} المقال المراد تقييمه: {essay_text} استنادًا إلى الأمثلة السابقة والمعايير المذكورة، قدم تقييمك بصيغة JSONعلى النحو التالي: {{"relevance": <"رقم من 0 إلى 2>, "organization": <"رقم من 0 إلى 5>, "vocabulary": <"رقم من 0 إلى 5>, "style": <"رقم من 0 إلى 5>, "development": <"رقم من 0 إلى 5>, "mechanics": <"رقم من 0 إلى 5>, "grammar": <"رقم من 0 إلى 5>}	You are an expert in assessing Arabic essays. Evaluate essays as a teacher would, based on the following weighted criteria: **Detailed Evaluation Criteria and Maximum Scores:** **1-Relevance** (Max: 2 points) Degree to which the essay addresses the assigned topic and covers its core aspects. **2-Organization** (Max: 5 points) Clear structure (introduction, coherent paragraphs, strong conclusion). Logical flow of ideas and smooth transitions between paragraphs. **3-Vocabulary** (Max: 5 points) Richness and accuracy of word choice. Use of varied, context-appropriate terms; avoidance of repetition. **4-Style** (Max: 5 points) Engaging and clear writing style. Correct and effective punctuation. Tone/style suited to the essay's purpose and audience. **5-Development** (Max: 5 points) Depth and elaboration of ideas. Strong arguments with supporting evidence (examples, citations) based on essay type (expository/persuasive). Multifaceted analysis of the topic. **6-Mechanics** (Max: 5 points) Spelling and punctuation accuracy. Proper overall formatting. **7-Grammar** (Max: 5 points) Correct syntax, morphology, and grammar. Proper sentence structure and clarity. Review the example essays: {examples_text} Evaluate the target essay against this prompt: {target_prompt_text} Assess the following essay: {essay_text} **Submit your evaluation in JSON format:** { "relevance": <0–2>, "organization": <0–5>, "vocabulary": <0–5>, "style": <0–5>, "development": <0–5>, "mechanics": <0–5>, "grammar": <0–5>, "total_score": <sum of all criteria>, "feedback": "<concise strengths/weaknesses per criterion>" }

Figure 2: Structured Prompt Template Applied in Task B's Automated Essay Scoring System.

A.2 Bias and Performance Visualizations for Tasks A and B

Figure 3 presents a bias visualization comparing human and model holistic scores for Task A. Figure 4 shows a trait-specific bias heat map for Task B, illustrating the difference between model and human scores. Figure 5 displays confusion matrices for the testing set's holistic score prediction for Task A, with Figure 5a specifically for Prompt 9 and Figure 5b for Prompt 10.

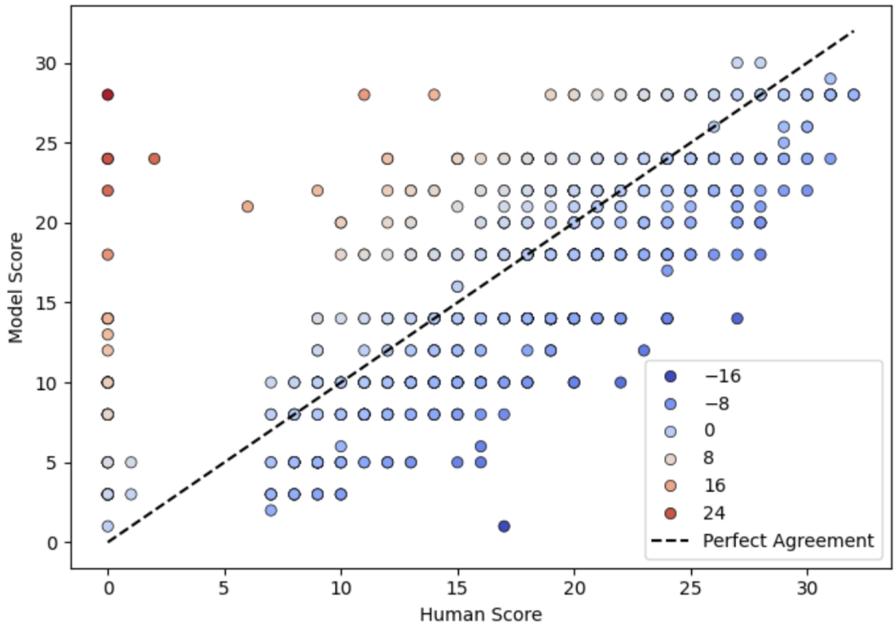


Figure 3: Bias Visualization: Human vs. Model Holistic Scores (Task A).

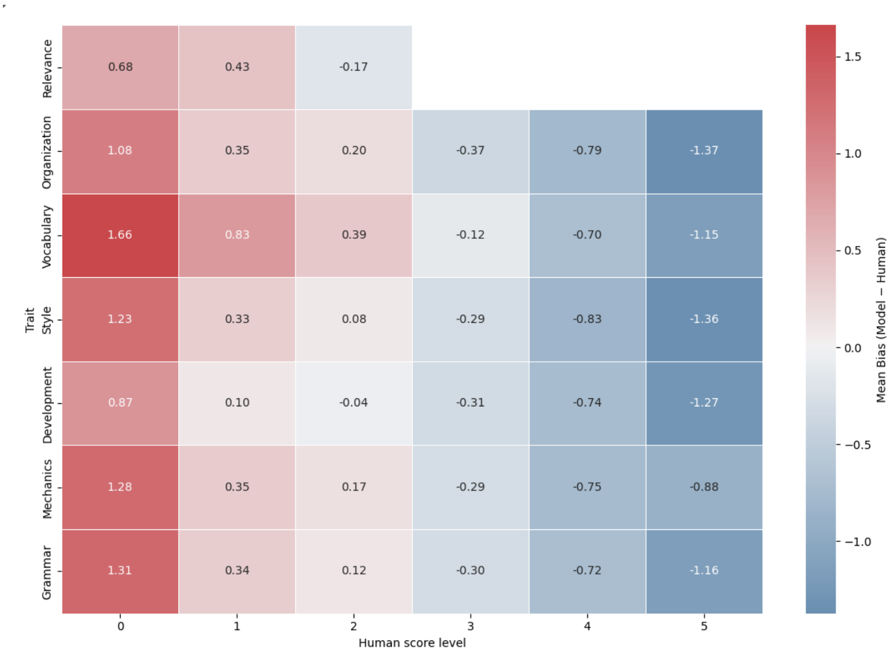
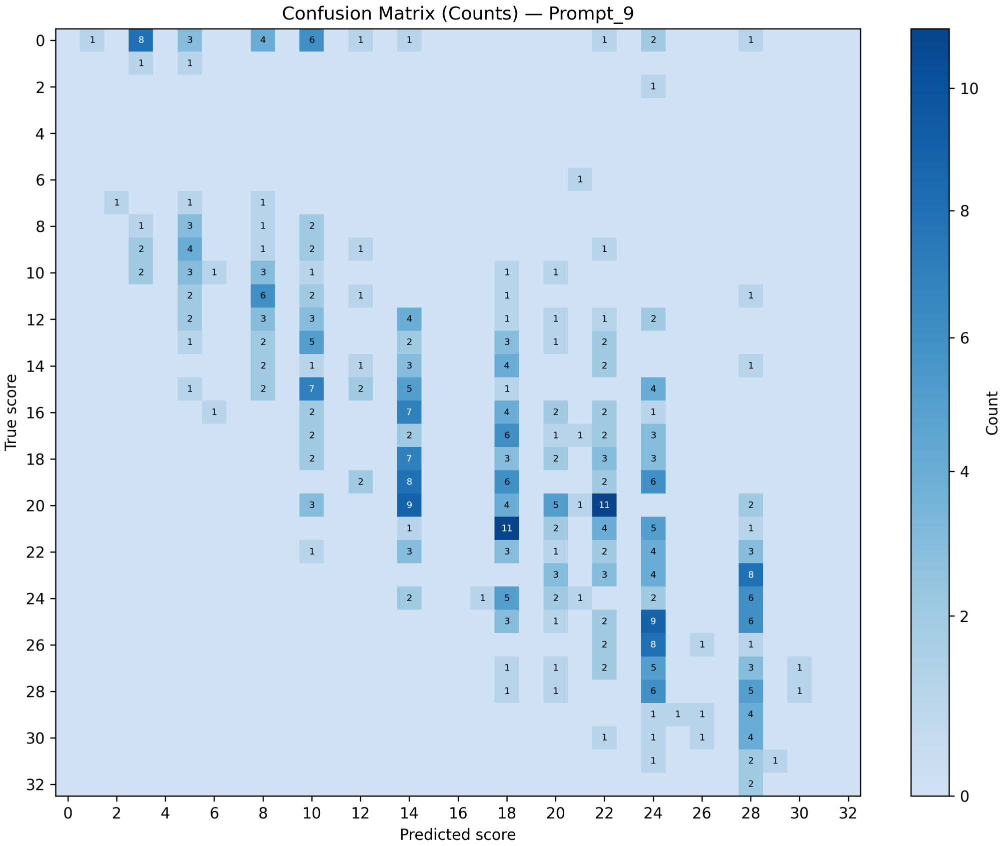
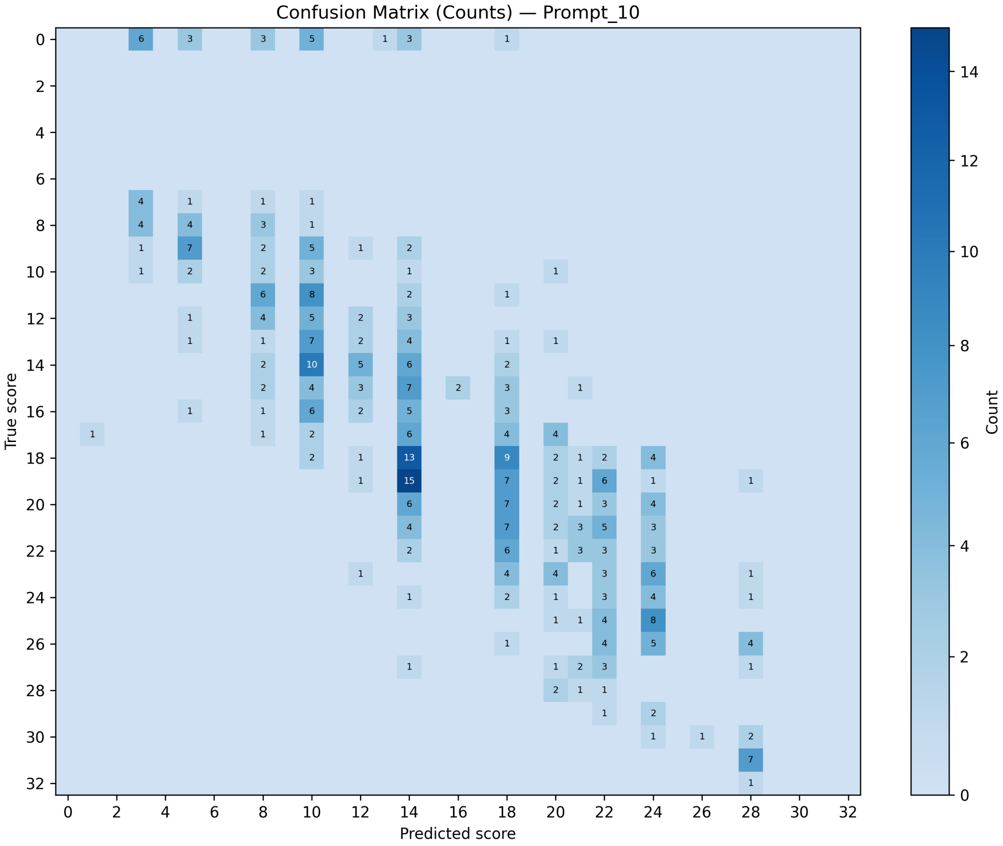


Figure 4: Task B Trait-Specific Bias Heat Map ($Model - Human$).



(a) Confusion Matrix for Testing set Holistic Score prediction For Task A: Prompt 9



(b) Confusion Matrix for Testing set Holistic Score prediction For Task A: Prompt 10

Figure 5: Confusion Matrix for Testing set Holistic Score prediction For Task A

MarsadLab at TAQEEM 2025: Prompt-Aware Lexicon-Enhanced Transformer for Arabic Automated Essay Scoring

Mabrouka Bessghaier[1], Md. Rafiul Biswas[2], Amira Dhouib[3], Wajdi Zaghouani[1]
[1]Northwestern University in Qatar, Qatar
[2]Hamad Bin Khalifa University, Qatar,
[3] LaTICE Lab, University of Kairouan
mbiswas@hbku.edu.qa
{mabrouka.bessghaier,wajdi.zaghouani}@northwestern.edu

Abstract

We present the MarsadLab submission to TAQEEM 2025 Shared Task A on Automated Essay Scoring (AES) in Arabic. Our system extends AraBERT with a prompt-type embedding and lexicon-based features. The lexicon captures statistical associations between word usage and essay quality under each prompt type, providing prompt-aware, interpretable signals that complement semantic embeddings. Our system achieved an average QWK of 0.438, highlighting both the promise and the challenges of incorporating prompt-sensitive lexical knowledge into AES. This work represents a first attempt at leveraging a task-aware lexicon for Arabic AES, showing that lexical features provide educational value through interpretability but also require more sophisticated integration. Future improvements could combine these lexical indicators with discourse-, syntax-, and content-level features, as well as explore richer fusion strategies to better exploit their potential.

1 Introduction

Automated Essay Scoring (AES) aims to predict human-assigned scores for student essays, offering applications in large-scale assessment and educational feedback. While AES has been widely studied for English, progress in Arabic remains limited due to scarce datasets, morphological complexity, and diverse rhetorical styles.

The TAQEEM 2025 Shared Task introduces the first large-scale benchmark for Arabic AES (Bashendy et al., 2025), evaluating systems on holistic score prediction across two writing prompts: explanatory and persuasive. This dual requirement makes the task particularly challenging, as effective systems must capture not only semantic meaning but also prompt-specific discourse and stylistic features. Furthermore, our submission was evaluated under the cross-prompt setting, where systems must generalize across different prompts, further increasing task difficulty.

Our submission explores a hybrid design that integrates AraBERT semantic embeddings with lexicon-based features. The lexicon captures statistical correlations between words and essay scores within each prompt type, offering interpretability and potentially complementing contextual embeddings. Although our results did not surpass the baseline, the analysis provides valuable insights into the difficulties of feature fusion and the role of lexical cues in Arabic AES.

2 Background

TAQEEM 2025 Shared Task A focuses on Arabic AES, where the goal is to predict a continuous holistic score for essays written in response to specific prompts. Each essay is linked to a prompt text, a prompt type (either explanatory or persuasive), and a human-assigned holistic score ranging from 0 to 32.

The dataset is structured around three components: (i) prompts that define the writing task and its type, (ii) student essays written in response to these prompts, and (iii) holistic scores provided by human raters. All essays are written in Modern Standard Arabic (MSA), covering academic writing across the two genres. Each instance thus consists of the essay text, the prompt information, and a holistic score. As shown in table 1, the training dataset of this task includes 425 essays written in response to two different prompts: one explanatory (215 essays) and one persuasive (210 essays). Each essay is annotated with a holistic human score ranging from 2 to 31, indicating a broad spread of writing quality. The distribution of essays across prompts is relatively balanced, ensuring that models are exposed to both explanatory and persuasive writing.

Automated essay scoring has been heavily studied, with early approaches relying mainly on regres-

Proceedings of The Third Arabic Natural Language Processing Conference, pages 998–1002
November 8-9, 2025 ©2025 Association for Computational Linguistics

sion models and hand-crafted linguistic features to capture aspects of writing quality. More recent research has increasingly focused on ensemble methods and deep learning models, which aim to better capture lexical, syntactic, and discourse-level characteristics of student writing (Ramnarain-Seetohul et al., 2025). However, AES has not advanced as rapidly due to linguistic complexity and the scarcity of large-scale annotated resources. One of the few early attempts is the work of Alqahtani (2019), who proposed a rule-based system for evaluating Arabic essays based on surface-level criteria such as spelling, punctuation, essay structure, coherence, and style (Alqahtani et al., 2019).

Several efforts have attempted to lay the groundwork for advancing AES in Arabic by providing resources that target key aspects of writing quality. For example, (Zaghouani et al., 2024) built the Qatari Corpus of Student Argumentative Writing. The proposed corpus presents a bilingual (Arabic/English) resource that captures discourse structure, coherence signals, and learner-writing phenomena. Complementary resources have focused on error annotation for learner Arabic, offering normalization and correction procedures, inter-annotator agreement metrics, and foundations for assessing grammar and fluency (Zaghouani et al., 2014). Similarly, gold-standard corrections for learner errors have been proposed in (Zaghouani et al., 2015), covering orthographic, morphological, syntactic, and punctuation mistakes, thereby enabling benchmarks for automated error correction and linguistic quality assessment. In addition, auxiliary resources such as Arabic diacritization guidelines provide conventions for orthography and phonology consistency, supporting disambiguation tasks relevant for spelling- and diacritic-aware quality assessment (Zaghouani et al., 2016). Research on punctuation and sentence-boundary annotation has also introduced resources for mechanics and readability, contributing cues for punctuation restoration and coherence modeling (Zaghouani and Awad, 2016). Together, these initiatives provide the linguistic and annotation foundations necessary for advancing Arabic AES, complementing scoring models by supplying resources on grammar, fluency, coherence, and overall writing quality.

In order to drive further progress in this domain, the TAQEEM 2025 Shared Task (Bashendy et al., 2025) presents the first extensive dataset for Arabic AES. Unlike previous small-scale or resource-specific efforts, it provides a balanced dataset of persuasive and explanatory essays for comprehensive scoring, allowing for systematic examination under cross-prompt settings.

In fact, the role of lexical features has been emphasized in assessing text quality. Such features describe the surface characteristics of textual responses, including single words, stemmed or lemmatized forms, prefixes, suffixes, or n-grams. Their extraction is relatively simple, and many algorithms have been proposed for Automatic Short Answer Grading (ASAG) tasks based on lexical similarity, overlap measures, or lexical statistics (Haller et al., 2022). These approaches laid an important foundation for later AES systems, especially in contexts where more sophisticated syntactic or semantic models were not available. In this work, we aim to further investigate the contribution of lexical features in the context of Arabic AES.

Prompt ID	Prompt Type	Essays
1	Explanatory	215
2	Persuasive	210
Total	–	425

Table 1: Distribution of essays and score ranges across prompts

3 System Overview

Our system extends a transformer-based regressor with a prompt-aware lexicon that captures lexical signals of essay quality. The overall workflow involves (i) building the lexicon from training data, (ii) extracting aggregated lexical features for each essay, and (iii) integrating these features with AraBERT embeddings in a hybrid architecture.

3.1 Task-Aware Lexicon Construction

We created a custom lexicon (1–3-grams) designed to reflect how word usage relates to essay quality under different prompt types (explanatory vs. persuasive). This process involved three main steps:

Merging resources. Essay texts, human-assigned scores, and prompt metadata were combined using shared identifiers (essay_id, prompt_id).

Preprocessing. Essays were normalized (removing diacritics and unifying variants of alif, ya, and taa marbuta), cleaned of non-Arabic characters, digits, and punctuation, and then tokenized into words.

Stopwords were deliberately retained, as function words such as connectives, discourse markers, and particles can vary systematically across explanatory and persuasive writing and thus provide useful discriminative signals. To avoid lexical leakage, any tokens appearing in the corresponding prompt text were excluded, ensuring the lexicon reflects only the language of student essays rather than the instructions.

Computing lexical statistics. For each unique `(word, prompt_type)` pair, we calculated:

- **Frequency:** how often the word occurs in essays of that prompt type.
- **Mean score:** average holistic score of essays containing the word.
- **Score variability:** the standard deviation of scores associated with the word.
- **Richness:** the number of unique score values linked to the word.
- **Z-score:** For each token, we compared the average score of essays containing that token with the mean score of all essays written under the same prompt type. The difference was normalized by the standard deviation of scores across the entire prompt type, yielding a classic z-score:

  $$z = \frac{\text{mean\_score(token)} - \text{mean\_score(prompt\_type)}}{\sigma_{prompt\_type} + 10^{-5}}$$

  This measures how far above or below the prompt-type average the token's essays tend to score, relative to the overall variability in that prompt type. Tokens occurring mainly in stronger essays have positive z-scores, while those associated with weaker essays receive negative z-scores.
- **Importance:** defined as frequency × |z-score|, highlighting words that are both frequent and strongly associated with higher or lower quality. defined as the logarithm of the token's document frequency, multiplied by its positive z-score:

  $$\text{importance} = \log(1 + \text{count}) \times \max(0, z)$$

  This formulation ensures that tokens are ranked higher when they are both frequent and associated with above-average essay scores, while logarithmic scaling prevents extremely common tokens from dominating the lexicon. Only tokens with positive importance were retained.

The result is a lexicon table where each row corresponds to a word conditioned on a prompt type, enriched with its statistical profile. This lexicon provides interpretable insight into vocabulary patterns rewarded or penalized by human raters.

3.2 Lexicon Feature Integration

While our lexicon construction relies on associations between words and essay scores, we do not assume that words directly cause higher or lower scores. Instead, certain lexical items tend to co-occur with patterns of stronger writing and can therefore serve as useful signals. In explanatory prompts, higher-scoring essays frequently include causal and elaborative markers such as الأسباب (“the reasons”), بشكل (“in a way”), أهم (“most important”), which help writers clarify causes, emphasize significance, or indicate conditions. In persuasive prompts, stronger essays often use العديد (“many”) to generalize claims,إلا (“except/but”) to introduce concessions or contrasts, and مما (“which/thereby”) to connect evidence with conclusions. These examples illustrate that while no single word determines essay quality, their systematic distribution provides interpretable clues about how students construct explanations or persuasive arguments. The task-aware lexicon is thus employed not as a causal determinant of scores but as a descriptive resource that highlights lexical tendencies associated with stronger or weaker essays under different prompt types. Such words can be markers of reasoning and structure, and their use often reflects the essay's quality. So the created lexicon was used to derive numerical features for each essay: **(i) Total importance:** the cumulative weight of all matched tokens in an essay. This reflects how much the essay overall makes use of words that are associated with higher importance scores. **(ii) Maximum importance** is the highest importance value among the essay's matched tokens, capturing the strongest single lexical signal present. **(iii) Average z-score (weighted)** specifies the central tendency of lexical associations in the essay, computed as the importance-weighted mean of token z-scores. These features were appended as auxiliary variables to each essay instance.

System	Prompt	QWK	MSE	RMSE	Avg. QWK	Avg. RMSE
Baseline	9	0.608	33.148	5.76	0.639	5.37
	10	0.670	24.862	4.99		
MarsadLab	9	0.447	40.431	6.36	0.438	7.07
	10	0.428	60.679	7.79		

Table 2: Comparison of Baseline and MarsadLab submissions

3.3 Model Architecture

We extended AraBERT-v2 with an additional branch for lexicon-based features, building a hybrid architecture that combines deep contextual embeddings with interpretable lexical signals. The system is based on the following steps:

1. **Essay encoding with AraBERT.** The essay text is encoded using AraBERT-v2 (encoder-only).

2. **Prompt-type signal.** A learned embedding representing the prompt type is added element-wise to the pooled essay vector. This provides the model with an explicit indication of whether the essay is explanatory or persuasive, helping it adapt its representations to genre-specific expectations.

3. **Lexicon feature extraction.** In parallel, each essay is mapped to a three-dimensional vector derived from the lexicon: (i) total importance, (ii) maximum importance, and (iii) weighted average z-score.

4. **Feature concatenation.** The pooled AraBERT vector (dimension 768, after prompt-type addition) is concatenated with the lexicon feature vector (dimension 3), yielding a combined representation of size 771. This joint representation ensures that both semantic and lexical signals are captured in a shared feature space.

5. **Regression head.** The combined vector is passed through a projection block consisting of a linear transformation, layer normalization, and dropout. A final linear layer produces a single logit, which is mapped to the valid score range $[0, 32]$ using a sigmoid and affine scaling. Training is optimized with Mean Squared Error (MSE) loss against the human-provided holistic scores.

This design allows the model to capture both deep semantic information (through AraBERT) and prompt-sensitive lexical cues (through the lexicon features). The concatenation step explicitly fuses these two types of signals, ensuring that the model considers not only meaning and discourse but also interpretable markers of explanation or persuasion that human raters often reward.

4 Experimental Setup

We trained models using AraBERTv2 with AdamW optimizer (learning rate 2e-5), batch size 8, max length 512, and early stopping on dev QWK. Evaluation follows official test protocol with QWK as the primary metric and RMSE as a secondary metric.

5 Results

Table 2 compares our submissions with the official baseline. The baseline achieved an average QWK of 0.639 (RMSE = 5.37), with consistent performance across both prompts. In contrast, our system obtained an average QWK of 0.438 (RMSE = 7.07). The drop was observed across both expository (Prompt 9) and persuasive (Prompt 10) essays. A likely reason for underperformance is the simplistic concatenation of features with AraBERT embeddings, which may not allow the model to weigh contextual versus lexical information dynamically. Another factor may be the small size of the dataset, which restricts the coverage of the constructed lexicon.

Compared to the baseline, our system underperformed in both QWK and RMSE. While the baseline achieved higher agreement with human raters, our hybrid AraBERT+lexicon approach demonstrated stable but lower performance. This suggests that our current fusion strategy does not fully exploit the complementary strengths of contextual and lexical features. Future work should explore attention-based fusion or prompt-adaptive weighting.

6 Conclusion

We presented the MarsadLab system for TAQEEM 2025 Task A, extending AraBERT with a prompt-type embedding and a task-aware lexicon for Arabic AES. The lexicon offered interpretable features—total importance, maximum importance, and weighted average z-score—that capture prompt-sensitive lexical tendencies. Our system achieved an average QWK of 0.438, showing that lexical features can be successfully integrated into AES, but also highlighting the need for more advanced methods to fully exploit their potential.

While the lexicon provides transparency and insight into genre-sensitive vocabulary, it remains correlational and incomplete. Future work should expand the lexicon across more prompts, combine it with discourse- and syntax-level features, and explore richer integration strategies such as attention-based fusion or prompt-adaptive regression.

7 Limitations

The task-aware lexicon we created gives useful and interpretable signals for essay scoring, but it is not enough on its own to capture the full complexity of writing quality. It reflects correlations between words and scores, yet essay quality also depends on broader aspects such as coherence, organization, and depth of reasoning, which cannot be reduced to lexical patterns. Another limitation is that the lexicon was built from only two prompts, one explanatory and one persuasive. This means some of the word associations may be domain-specific and tied to the topics of these prompts rather than general markers of writing quality. Finally, the way we integrated lexical features with AraBERT relied on simple concatenation, which likely limited the model's ability to make effective use of both contextual and lexical information. These points show that while the lexicon is a helpful resource, it should be seen as a first step. Future work should expand it to more prompts, add discourse- and syntax-level features, and test more advanced fusion methods to improve both generality and performance.

Acknowledgements

This study was supported by the grant NPRP14C-0916-210015, awarded by the Qatar Research, Development and Innovation Council (QRDI).

References

Hmoud Alqahtani, Sabri Mahmoud, and Shadi Al-Saqqa. 2019. Automated essay scoring for arabic essays using content and text features. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–6.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Walid Massoud, Houda Bouamor, and Tamer Elsayed. 2025. TAQEEM 2025: Overview of the First Shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

Stefan Haller, Adina Aldea, Christin Seifert, and Nicola Strisciuglio. 2022. Survey on automated short answer grading with deep learning: from word embeddings to transformers. *arXiv preprint arXiv:2204.03503*.

Vidasha Ramnarain-Seetohul, Yasmine Rosunally, and Vandana Bassoo. 2025. Ensemble and hybrid models in automated essay scoring: A literature review. *SN Computer Science*, 6(6):729.

Wajdi Zaghouani, Abdelhamid Ahmed, Xiao Zhang, and Lameya Rezk. 2024. QCAW 1.0: Building a qatari corpus of student argumentative writing. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13382–13394, Torino, Italia. ELRA and ICCL.

Wajdi Zaghouani and Dana Awad. 2016. Building an arabic punctuated corpus. 2016(1).

Wajdi Zaghouani, Houda Bouamor, Abdelati Hawwari, Mona Diab, Ossama Obeid, Mahmoud Ghoneim, Sawsan Alqahtani, and Kemal Oflazer. 2016. Guidelines and framework for a large scale Arabic diacritized corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3637–3643, Portorož, Slovenia. European Language Resources Association (ELRA).

Wajdi Zaghouani, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Heider, and Kemal Oflazer. 2015. Correction annotation for non-native Arabic texts: Guidelines and corpus. In *Proceedings of the 9th Linguistic Annotation Workshop*, pages 129–139, Denver, Colorado, USA. Association for Computational Linguistics.

Wajdi Zaghouani, Nizar Habash, Behrang Mohit, Abeer Heider, Alla Rozovskaya, and Kemal Oflazer. 2014. Annotation guidelines for non-native arabic text in the qatar arabic language bank. 2014(1).

Author Index